WAGNER, PAUL BARNABY

AN APPROACH TO CUSTOM TYPE-CURVE GENERATION FOR PRESSURE TRANSIENT ANALYSIS

UNIVERSITY OF ALASKA M.S. 1983

AN APPROACH TO CUSTOM TYPE-CURVE GENERATION FOR PRESSURE TRANSIENT ANALYSIS

A

THESIS

Presented to the Faculty of the University of Alaska

in Partial Fulfillment of the Requirements

for the Degree of

MASTER OF SCIENCE

By

Paul Barnaby Wagner, B.A.

Fairbanks, Alaska

December, 1983

AN APPROACH TO CUSTOM TYPE-CURVE

GENERATION FOR PRESSURE TRANSIENT ANALYSIS

RECOMMENDED:

Chairman, Advisory Committee

Department Head

APPROVED:

Vice-Chancellor for Research and Advanced Study

25 October 1983

Date

ABSTRACT

Historically, the introduction of the source function method greatly simplified the derivation of analytic solutions for a variety of reservoir flow geometries. Solutions formulated in this way are not directly comparable because the definitions of the variables used in the past to classify solution behavior varied. Additionally, a computational problem exists when certain of the source functions are used at early values of time.

For this work, a generalized set of dimensionless variables was defined. A method is presented which makes it simple to formulate analytic solutions for a wide variety of well drainage geometries. A complete derivation of source functions stable for early-time calculations is presented. A method of combining these new source functions with the older ones is explained. A computer program is presented which implements these formulations automatically enabling a form of computer-aided type-curve matching and extending the practical accessibility to these solutions. An illustration of these computer-aided methods is given for two examples, one being an alternate interpretation of a published analysis, and the other being an analysis of the previously unpublished case of a fractured well in an elongated linear flow system.

The methods are illustrated in two example applications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

The author would like to express his sincere appreciation to Dr. Christine Ehlig-Economides and Dr. Michael J. Economides for their support and constructive criticism; to Dr. Christine Ehlig-Economides, Dr. Michael J. Economides, and Dr. Russell D. Ostermann of the University of Alaska, Department of Petroleum Engineering, for serving on his committee; and to Beth Bedwell, for her patience in typing and editing this document.

INTRODUCTION

Requirements for detailed information about the formation, in order to produce more oil from existing reservoirs, result in the need for in-depth analysis. This analysis is complicated by the multitude of pressure solutions (type-curves) that have been published. If the geometry of the well and its drainage volume are not fully described, one must sort through a large body of information in order to find a solution which properly models the basic geometry present. If the geometry is known, often the solutions published for that geometry are for idealized well placements and boundary shapes, and thus a reasonable estimate is possible, but not necessarily an exact match. Clearly, any technique which can make the existing solutions and geometries more accessible would be of great benefit in the analysis of a well test.

Gringarten and Ramey (1973) generalized the process for deriving pressure solutions for a wide variety of geometries by introducing the practical application of source and Green's functions to unsteady flow problems. They illustrated the product method of separating a complicated, three-dimensional reservoir geometry into two or three simpler one-dimensional geometries, each of which can be solved using the source function method. This paper was the basis for the solutions given in a great number of later papers by many other authors.

After the introduction of source functions, Gringarten, Ramey, and Raghavan (1974) developed solutions for infinite-conductivity and uniform flux fractures. The fracture dimensions could be made large or small, compared to the dimensions of the drainage volume for the well, so that both the case of an infinite system and the case of a bounded system could be modelled. The case of a vertical fracture was previously studied by Prats (1961) and Prats, et al. (1962) for a circular system with a constant-pressure outer boundary, using elliptical calculus; by deSwann (1976) for the infinite case only, using Laplace transformation; and by Russell and Truitt (1964) for a rectangular system with no-flow boundaries, using finite-difference techniques. Raghavan and Hadinoto (1978) used source functions when they extended the infinite-conductivity and uniform flux vertical fracture case by considering the outer boundaries of the reservoir as being constant-pressure, as they would be in the case of a reservoir with an aquifer bouunding the oil from below. Gringarten (1978) used source functions when describing the case of reservoir limit testing. Cinco, et al. (1975) used source functions to derive solutions for an inclined fracture. Raghavan, et al. (1978) used them to study the effect of vertical fracture penetration on pressure and on productivity. Buhidma and Raghavan (1980) modelled the effect of a bottomwater drive on a partially-penetrating well, an extension of the model with a fully-penetrating well. In summary, a great many useful

geometries were simulated using the source function method, and, as implied by Gringarten and Ramey (1973), a very large number of situations are available for solution in this manner.

Cinco, et al. (1979) and Economides, et al. (1980) used the source function approach to study the pressure response for a fractured well in a geothermal reservoir shaped like a parallelepiped, with a constant-pressure boundary beneath the fracture at some unknown depth, which represented a hypothetical boiling interface in a steam reservoir. Introduced in these papers were new source functions which were more easily calculated for early-time analysis than the Gringarten and Ramey source functions. A general method of deriving these early-time solutions has not been published.

In this work, the use of source functions will be generalized further by choosing a consistent set of dimensionless variables to use for any problem solvable by the source function method. This consistency in notation can be exploited to allow one to construct a pressure solution for any chosen geometry with a minimum of effort. In fact, it will be shown that this method is completely deterministic, making possible the translation of these ideas into a form of computer-aided type-curve matching.

The purpose of this study is three-fold. First, the derivation and behavior of the early-time source functions mentioned above is presented. Second, a consistent definition of dimensionless variables is presented. Lastly, a method of combining the early-time source functions with their previously-published counterparts, and a method

of utilizing these combinations along with the rest of the published source functions in an automatic fashion, is presented. It will be shown how this automatic method can be used to simulate the same geometries mentioned above, and how one can model many cases not previously published with equal ease.

THEORY

In order to develop the mathematical theory behind the results to be presented, this chapter has been broken down into four areas. The rationale behind the differential equations used is developed in the first section. Next, the decomposition of the basic problem into smaller, simpler problems is discussed. Next, the theory of Green's and source functions is presented as a method of solving the simpler problems. And finally, the method of superposition is applied to some of the source functions presented, and the benefits of its application are established. The application of this theory to an actual situation is left to the following chapters.

A. The Diffusivity Equation.

The fundamental differential equation for representing pressure in a porous medium as a function of position and time is the diffusivity equation. The schematic diagram shown in Fig. 2.1 illustrates mathematical concepts which will facilitate the following discussion of the diffusivity equation and the systems of equations to be solved in this work.

Flow of a slightly compressible fluid at an observation point $\bar{x}$ within the porous and permeable media contained in the domain D and bounded by S_e is modelled using a differential equation which is derived by an analysis of differential elements using the

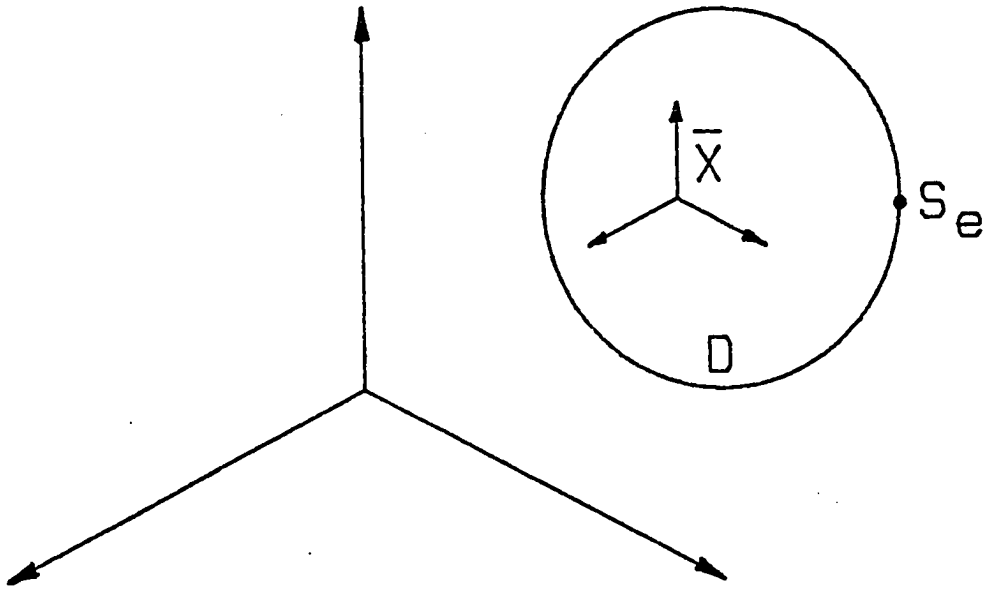


Fig. 2.1. Schematic diagram of reservoir domain.

continuity equation (conservation of mass) and Darcy's law. By assuming (1) constant permeability, porosity, and fluid viscosity over all time and space; (2) small pressure gradients near all points in D at all times; and (3) negligible gravity effects, the diffusivity equation can be written as:

$$\nabla^2 p = \frac{1}{\eta} \frac{\partial p}{\partial t} \tag{2.1}$$

where η is the diffusivity constant, defined by:

$$\eta = \frac{k}{\phi \mu c} \tag{2.2}$$

By applying the definition of the differentiation operator (∇) for any given coordinate system, the diffusivity equation in that system can be written. Hence, for a three-dimensional cartesian system, the diffusivity equation is:

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} = \frac{1}{\eta} \frac{\partial p}{\partial t} \tag{2.3}$$

and for a cylindrical system with coordinates for radial distance and elevation, the diffusivity equation is:

$$\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial p}{\partial r} \right) + \frac{\partial^2 p}{\partial z^2} = \frac{1}{\eta} \frac{\partial p}{\partial t} \tag{2.4}$$

When boundary conditions appropriate to a specific reservoir geometry are assigned to all points of S_e , a system of equations result which must then be solved for the unknown $p(\bar{x},t)$.

As was pointed out in the Introduction, solutions for many practical systems appear throughout the petroleum engineering literature. Earlier, Carslaw and Jaeger (1946) demonstrated at length the use of the point source solution of Lord Kelvin (1884) in solving similar problems in heat conduction. The concept of the point source is essential to the understanding of the work that is to be presented here. The point source solution of Lord Kelvin is the result of solving the diffusivity equation in the spherical (one-dimensional) coordinate system. It was assumed that all flow of heat (or fluid, in the discussion to follow) was from a zero-dimensional point located within the domain D. The surface S_e was assumed to be infinitely distant, and the temperature where the radial distance approached infinity was zero. One of two conditions were stated for the source point: either a constant flux of heat (fluid) emanated to or from the source point (the Neumann problem), or the temperature (pressure) of the source point was fixed (the Dirichlet problem). Lord Kelvin derived a solution for the Neumann problem, which is the type of problem dealt with in this work.

For reservoir flow problems, a corresponding form of the point source solution was derived by Nisle (1958), and is given by:

$$\Delta p = \frac{q(t)}{8\phi c(\pi\eta t)^{3/2}} \exp\left[\frac{-\bar{x}\bar{x}_s^2}{4\eta t}\right] \qquad (2.5)$$

Where Δp is the pressure change at $\bar{x}$ at time t, and $\bar{\bar{x}}\bar{\bar{x}}_s$ represents the distance between the point $\bar{x}$ and the point $\bar{x}_s$, where a point source of strength q creates an instantaneous flow disturbance in a reservoir assumed to be infinite in extent. The quantity q(t) is the volume of fluid produced or injected at the time t. For fluid flow over any period of time t, Eq. 2.5 must be integrated over the time interval from 0 to t:

$$\Delta p(\bar{x},t) = \frac{1}{8\phi c(\pi\eta)^{3/2}} \int_0^t \frac{q(\tau)}{(t-\tau)^{3/2}} \exp\left[\frac{-\bar{\bar{x}}\bar{\bar{x}}_s^{\,2}}{4\eta(t-\tau)}\right] d\tau \quad (2.6)$$

Gringarten, et al. (1973) showed that the pressure change due to fluid flow to or from a continuous source of finite dimensions is found by integrating over the source volume D_w:

$$\Delta p(\bar{x},t) = \frac{1}{8\phi c(\pi\eta)^{3/2}} \int_0^t \frac{q(\tau)}{_s(t-\tau)^{3/2}} \cdot$$

$$\int_{V_s} \exp\left[\frac{-\bar{\bar{x}}\bar{\bar{x}}_s^{\,2}}{4\eta(t-\tau)}\right] d[V_s]d\tau \quad (2.7)$$

Because the distance $\bar{\bar{x}}\bar{\bar{x}}_s^2$ can be written as the sum of the squares of the distances in each dimension (true for both cylindrical and cartesian geometries), an equivalent form of Eq. 2.6 can be written for the cartesian system as:

$$\Delta p(\bar{\bar{x}},t) = \frac{1}{8\phi c(\pi\eta)^{3/2}} \int_0^t \frac{q(\tau)}{(t-\tau)^{3/2}} \cdot$$

$$\exp\left[\frac{-d_x^2}{4\eta(t-\tau)}\right] \exp\left[\frac{-d_y^2}{4\eta(t-\tau)}\right] \exp\left[\frac{-d_z^2}{4\eta(t-\tau)}\right] d\tau \qquad (2.8)$$

This concept of distributing the solution among the coordinates is crucial to the development of the derivations to follow.

B. Decomposition of a Multidimensional Problem.

The solution of Eq. 2.3 for arbitrary boundary conditions can be especially difficult. However, if the boundary conditions can be written separately for each of the spatial dimensions, the problem can be decomposed into three one-dimensional problems, each of which can be handled by more conventional means. To show this, assume that Eq. 2.3 can be written as:

$$\sum_{j=1}^{3} \frac{\partial^2 \bar{p}}{\partial x_j} = \frac{1}{\eta} \frac{\partial \bar{p}}{\partial t} \qquad (2.9)$$

where x_j is any one of the three space variables. If the solution $\bar{p}$ can be written as:

$$\bar{p} = p_1 \cdot p_2 \cdot p_3 \qquad (2.10)$$

where p_j is the solution for the dimension of x_j, then, by substituting Eq. 2.10 into Eq. 2.9:

$$p_2 p_3 \frac{\partial^2 p_1}{\partial x_1^2} + p_1 p_3 \frac{\partial^2 p_2}{\partial x_2^2} + p_1 p_2 \frac{\partial^2 p_3}{\partial x_3^2}$$

$$= \frac{1}{\eta} \left(p_2 p_3 \frac{\partial p_1}{\partial t} + p_1 p_3 \frac{\partial p_2}{\partial t} + p_1 p_2 \frac{\partial p_3}{\partial t} \right) \qquad (2.11)$$

By regrouping terms, three separate equations result:

$$p_2 p_3 \frac{\partial^2 p_1}{\partial x_1^2} = \frac{1}{\eta} p_2 p_3 \frac{\partial p_1}{\partial t}$$

$$p_1 p_3 \frac{\partial^2 p_2}{\partial x_2^2} = \frac{1}{\eta} p_1 p_3 \frac{\partial p_2}{\partial t}$$

$$p_1 p_2 \frac{\partial^2 p_3}{\partial x_3^2} = \frac{1}{\eta} p_1 p_2 \frac{\partial p_3}{\partial t} \qquad (2.12)$$

Dividing each of the above three equations by the pressure solutions for the other two dimensions leaves three equations identical in form to the linear diffusivity equation. When each of these equations is solved, with the boundary conditions imposed on each identical to the ones which were desired to be applied to the original equation, then Eq. 2.9 is solved, and the solution is given by Eq. 2.10 via the solutions obtained by solving each equation in Eq. 2.12. This powerful process is termed in mathematics as "separation of variables".

The boundary conditions applied to each dimension must be chosen carefully, because the intersection of the separate boundary conditions give the boundary conditions for the solution, just as the product of the individual, one-dimensional pressure solutions gives the solution itself. It is useful at this point to discuss the

distinction between the "inner" and "outer" boundary conditions. In this work, the inner boundary is made up of a continuous volume of point sources contained in D_w, the volume being referred to as V_s. The outer boundary is the surface S_e, the surface bounding the domain D.

In practice, the inner boundary condition is used to model a well, and the outer boundary condition applies to the conditions at the boundary of the drainage volume of the well. The following example demonstrates how these concepts are applied.

EXAMPLE 2.1

In this example, the methods in this section are used to derive the boundary conditions needed to derive the pressure solution for the system shown in Fig. 2.2. The inner boundary consists of the line, which represents a flowing well; the outer boundary is a rectangular parallelepiped. By considering the outer boundary impermeable, this models the boundary of the drainage volume for a well in a fully-developed field where there are other wells surrounding the well in a rectangular pattern. The problem can be decomposed into three separate problems, one for each dimension, as shown in Fig. 2.3. The domains of the source in the x and y dimensions are equivalent to planes, which intersect to form the line. The domain of the source in the z-direction is a slab which completely occupies the space inside the outer boundaries in the z-dimension. The outer boundaries for all

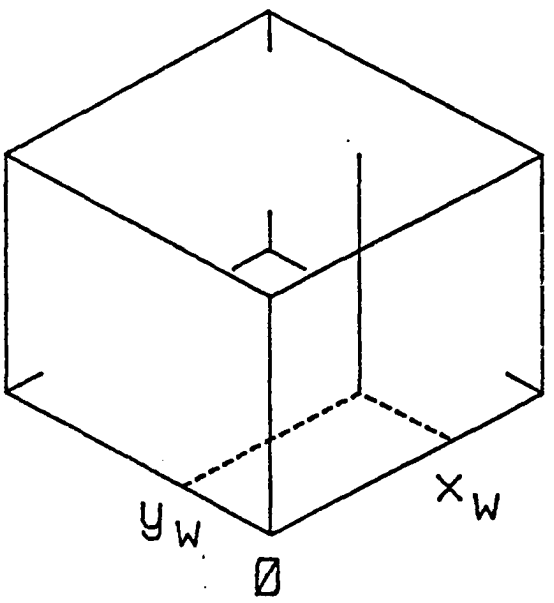


Fig. 2.2. Line source well in a rectangular reservoir.

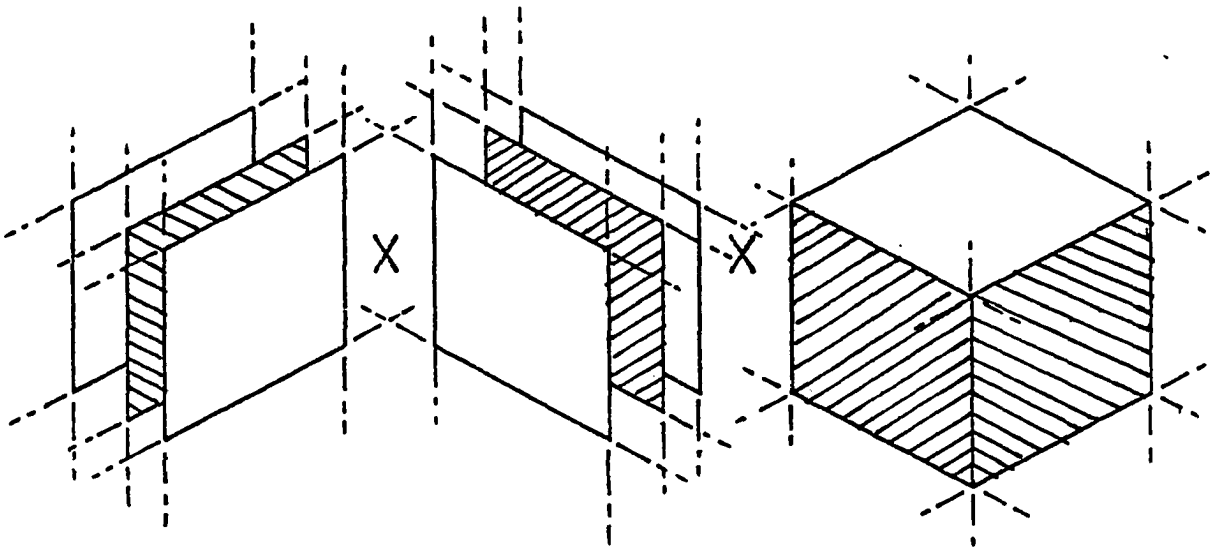


Fig. 2.3. Intersection of sources and boundaries (sources shaded).

three dimensions are impermeable, or no-flow boundaries. The solution of the whole problem derives from the solution of the problems for each of the three dimensions, where the problem for any given dimension has both the inner and outer boundary conditions specified for it. A straightforward method of solving the problem for each dimension is presented in the next section. Hence, a solution for this problem will be provided later. The sources referred to in the above example will be referred to in later sections as extended sources, i.e. sources which are no longer just a point, but a collection of points. Extended sources intersect to form sources of finite volume such as an area or length. The "line source" in the example is a length, resulting from the intersection of one extended slab source (which provides the length) and two planar sources.

In this work, the one-dimensional geometries discussed are the linear, cylinder surface, and solid cylinder sources without outer boundaries; the planar source with and without outer boundaries; and the slab source with and without outer boundaries. The first set of geometries are increasingly complex representations of the wellbore, the second set is necessary for creating planar objects such as fractures, and the last set gives a finite, non-zero dimension to any source it intersects with. This would, for example, allow the radial wells to be fully- or partially-penetrating, and would give a fracture a finite length.

The single-dimensional problems of Eq. 2.12 can be solved in a number of ways. One convenient way is to use the powerful Green's

function approach, described in the next section. Alternative forms of Green's function will be discussed in the final section of this chapter.

C. Source and Green's Functions.

The Green's function approach to solving the diffusivity equation with boundary conditions imposed is well known for heat conduction problems. Detailed theoretical derivations for the application of this approach to flow problems are given in Carslaw and Jaeger (1946). The following summary is adapted from Gringarten and Ramey (1973).

The solution of Eq. 2.1 is uniquely determined by the initial and boundary conditions. These are: (1) the initial pressure distribution in D; (2) either the flux across S_e for all times (Neumann), or the value of the pressure everywhere on S_e for all times (Dirichlet); and (3) the flux generated by the source. The constituents of condition (2) can be mixed if needed. The Green's function for D is defined as: the pressure change that is created at the point $\bar{x}$ at the time $t > \tau$ by an instantaneous source of strength unity at the point $\bar{x}_s$ which began constant flux at the time $t = \tau$. D is at a static equilibrium pressure initially, and S_e is either impermeable to flow or is at zero pressure everywhere. This function, symbolized as $G(\bar{x}, \bar{x}_s, t-\tau)$, is related to the solution by:

$$\Delta p(\bar{x},t) = \int_0^t \int_{S_e} \frac{\partial}{\partial n} G(\bar{x},\bar{x}_s,t-\tau)dS_e d\tau \qquad (2.13)$$

where $\partial/\partial n$ denotes an outward-drawn ormal derivative. The function has the following properties:

1. It is a solution of the adjoint diffusivity equation. For two differentiation operators L and M, if $L(\nu)$ represents the diffusivity equation, $M(\mu)$ is the adjoint form defined by requiring that the expression $\mu L(\nu) - \nu M(\mu)$ be integrable. In this case, $L \doteq [\eta\nabla^2 - \frac{\partial}{\partial t}]$ and $M \doteq [\eta\nabla^2 + \frac{\partial}{\partial t}]$ for $t > \tau$.(μ and ν in this case represent the pressure function, μ is not the fluid viscosity).

2. It is symmetrical in the two points $\bar{x}$ and $\bar{x}_s$.

3. It is a delta function. It is similar to the Dirac delta function in that it vanishes everywhere except at $\bar{x}_s$, where it becomes infinite in such a way that it remains continuous and integrable. Therefore, for any function $f(\bar{x})$:

$$\lim_{t\to\tau} \int_D f(\bar{x})G(\bar{x},\bar{x}_s,t-\tau)d\bar{x} = f(\bar{x}_s) \qquad (2.14)$$

In particular, for $f(\bar{x}) = 1$, the identity:

$$\lim_{t\to\tau} \int_D G(\bar{x},\bar{x}_s,t-\tau) = 1 \qquad (2.15)$$

is true for all times $\tau \geq 0$.

4. On the outer boundary: (a) if the pressure is prescribed for S_e, the Green's function vanishes for x_s on S_e; (b) if the flux is prescribed on S_e, the normal derivative of it (as given in Eq. 2.13) vanishes for x_s on S_e; (c) if D is infinite in extent, it is zero when x_s is at infinity.

The Green's function thus specified is the same for any shape of the source, as long as the source is established to be some particular type. This type must be either one of constant flux or of constant pressure. Only the constant flux case is considered here.

If D_w is the domain of an extended source, and $\bar{x}_w$ is some point in the source, the pressure at $\bar{x}$ at time t for an initial pressure distribution $p_i(\bar{x})$ is:

$$\int_{D_w} p_i(\bar{x}_w)\, G\,(\bar{x},\bar{x}_w,t)d\bar{x}_w - p(\bar{x},t) =$$

$$\int_0^t \int_{D_w} q(\bar{x}_w,\tau)\, G\,(\bar{x},\bar{x}_w,t-\tau)d\bar{x}_w\, d\tau$$

$$-\eta \int_0^t \left[\int_{S_e} [G(\bar{x},\bar{x}_e,t-\tau)\, \frac{\partial p(\bar{x},\tau)}{\partial n(\bar{x})}\right.$$

$$\left. - p(\bar{x},\tau)\, \frac{\partial G(\bar{x},\bar{x}_e,t-\tau)}{\partial n(\bar{x})}] \, d[S_e] \right]_{\bar{x}_e \varepsilon S_e} d\tau \qquad (2.16)$$

If the initial pressure is uniform in D, the leftmost integral is p_i . If D is infinite, the second term of Eq. 2.16 drops out, since it only accounts for the outer boundary conditions. If S_e exists, and is of zero flux (no-flow boundary), $\frac{\partial p(\bar{x},\tau)}{\partial n(\bar{x})} = 0$. If S_e is of zero pressure, $p(\bar{x},t) = 0$. Thus, if these constraints are observed, the second term always drops out, and:

$$p_i - p(\bar{x},t) = \Delta p = \frac{1}{\phi c} \int_0^t \int_{D_w} q(\bar{x}_w,\tau) G(\bar{x},\bar{x}_w,t-\tau) d\bar{x}_w d\tau \qquad (2.17)$$

If flux is uniform over the source, then $q(\bar{x}_w,\tau) = q(\tau)$. This term represents the flux per unit volume of the source. Because the inner integral involves the source domain D_w only, it can be treated separately. In this way,

$$S(\bar{x},t-\tau) = \int_{D_w} G(\bar{x},\bar{x}_w,t-\tau) d\bar{x}_w \qquad (2.18)$$

is the uniform flux source function for D_w, and:

$$\Delta p = \frac{1}{\phi c} \int_0^t q(\tau)\, S\,(\bar{x},t-\tau) d\tau \qquad (2.19)$$

Finally, if the volume of the source is given, explicitly, and $q(\tau) = q$ for constant-rate production for the entire source,

$$\Delta p = \frac{q}{\phi c V_s} \int_0^t S(\bar{x}, t-\tau) d\tau \qquad (2.20)$$

The uniform flux source functions are also used to describe infinite conductivity conditions in a well (or in a fracture, if the well intercepts it). However, the corresponding pressure must be measured within the source at an appropriate point. The location of this point was discussed by Gringarten and Ramey (1974). The case of finite conductivity sources is beyond the scope of this work. Solutions for finite conductivity fractures include the work done by Cinco-L., Samaniego-V., and Dominquez-A. (1978). In their work, discretization of the finite-conductivity fracture was used (the fracture was divided into 20 segments, and time was discretized into 10 intervals per log cycle). Because the finite conductivity case does not seem to be amenable to analytic solution, this case was not attempted.

By using the concept of decomposition discussed earlier, a source function can be decomposed into source functions for each dimension. In this way, a one-dimensional Green's function is sought, which is then integrated to give a source function as in Eq. 2.18, and is then substituted into Eq. 2.20 to give a one-dimensional pressure

solution. Therefore, the product of source functions is equivalent to the product of one-dimensional solutions, when this product is integrated over time as in Eq. 2.20.

A table of source functions for various one-dimensional boundary conditions was presented by Gringarten and Ramey (1973), and is given here as Table A.1. in the Appendix. The extended source, or slab source, is created by integrating the plane source over a distance:

$$II(x) = \int_{x_w - x_{f/2}}^{x_w + x_{f/2}} \{\exp[\frac{-(x-x'_w)^2}{4\eta\tau}]/2\sqrt{\pi\eta\tau}\}\, dx'_w$$

$$= \frac{1}{2}\left[\operatorname{erf}\left(\frac{x_{f/2} + (x-x_w)}{2\sqrt{\eta\tau}}\right) + \operatorname{erf}\left(\frac{x_{f/2} - (x-x_w)}{2\sqrt{\eta\tau}}\right)\right] \qquad (2.21)$$

where the term in brackets is the plane-source source function, I(x). III(r) is derived by using the product method:

$$III(r) = \{\exp(\frac{-(x-x_w)^2}{4\eta\tau})/2\sqrt{\pi\eta\tau}\} \cdot \{\exp(\frac{-(y-y_w)^2}{4\eta\tau})/2\sqrt{\pi\eta\tau}\}$$

$$= [\exp(\frac{-r^2}{4\eta\tau})/4\pi\eta\tau] \qquad (2.22)$$

where $r^2 = (x-x_s) + (y-y_s)^2$. IV(r) is derived by creating a surface of revolution, rotating III(x) about a point at $r = r_w$. Similarly, V(r) is derived by integrating IV(r) over a given range of radii. Finally, VI(d) can be derived by the product method, using three separate instances of I(x), one for each of the dimensions.

Gringarten and Ramey (1973) produced the bounded source functions VII(x) through XII(x) by converting into petroleum terms the analogous heat conduction forms presented by Carslaw and Jaeger (1946). These were derived by using Laplace transformation and inversion. The bounded sources, when multiplied together, produce bounded reservoirs, such as those resulting from fully-developed well patterns, and from reservoirs surrounded by water zones or sealing faults. The sources within these bounds can, for example, be representative of a well, or of a uniform flux (or infinite conductivity) fracture, as described above.

With source functions, the solution of the problem outlined in Example 2.1 is readily derived. The source function to use in Eq. 2.20 is the product of VII(x), VII(y) (y substituted for x), and X(z) (z substituted for x):

$$S(x,x_w,\tau) = \left[\frac{1}{x_e}\left(1 + 2 \sum_{n=1}^{\infty} \exp\left(\frac{-n^2\pi^2\eta\tau}{x_e^2}\right) \cos n\pi \frac{x_w}{x_e} \cos n\pi \frac{x}{x_e}\right)\right]$$

$$\cdot \left[\frac{1}{y_e}\left(1 + 2 \sum_{n=1}^{\infty} \exp\left(\frac{-n^2\pi^2\eta\tau}{y_e^2}\right) \cos n\pi \frac{y_w}{y_e} \cos n\pi \frac{y}{y_e}\right)\right] \cdot [1] \qquad (2.23)$$

$X(z) = 1$ in this case for a slab that fully penetrates the formation; this can be seen by substituting $x_w = 1/2\ x_e$ and $x_f = x_e$ into X(x) and noting that the sine and cosine cancel for all n. Thus, the final solution is:

$$\Delta p(\bar{x},t) = \frac{1}{\phi c} \int_0^t \frac{1}{x_e y_e}$$

$$\cdot \left[1 + 2 \sum_{n=1}^{\infty} \exp\left(\frac{-n^2\pi^2\eta\tau}{x_e^2}\right) \cos n\pi \frac{x_w}{x_e} \cos n\pi \frac{x}{x_e}\right]$$

$$\cdot \left[1 + 2 \sum_{n=1}^{\infty} \exp\left(\frac{-n^2\pi^2\eta\tau}{y_e^2}\right) \cos n\pi \frac{y_w}{y_e} \cos n\pi \frac{y}{y_e}\right] d\tau \quad (2.24)$$

Equivalent forms for the source functions with outer boundaries (VII(x) through XII(x)) can also be produced by the method of images, or superposition. This method, as applied to source functions, has not been fully discussed in the literature before.

D. Superposition for Source Functions.

Sources can be distributed through space to produce boundaries between them, just as image wells are distributed around the producing well to simulate drainage boundaries for the producing well. The source function solutions of Cinco, et al. (1979) and of Economides, et al. (1980), include particular cases of this procedure, but are given without discussion of their origin, and without discussion of the authors' claim that they are better for early values of time. A complete description of this method as it applies to source functions is presented for the first time here, along with a discussion of the behavior of these new functions with time.

To use superposition, first recall that the pressure distribution resulting from an extended source is given by Eq. 2.7, which is in effect the sum of all the pressure drops created by each of the point sources. In a similar way, the effect of one or more distinct extended sources can be summed. By pairing parallel sources of like or opposite sign and of equal magnitude, a planar boundary with a no-flow or constant pressure condition, respectively, results.

For example, if two identical plane sources are parallel, a flow boundary will be established between them. If the source functions of these two planes are added together, a no-flow boundary is created, as depicted in Fig. 2.4. Since the sources are implicity unit strength, the boundary forms exactly halfway between the plane sources. The source used to create this boundary is referred to as an image source, as if a reflection of the first source through the boundary plane. If

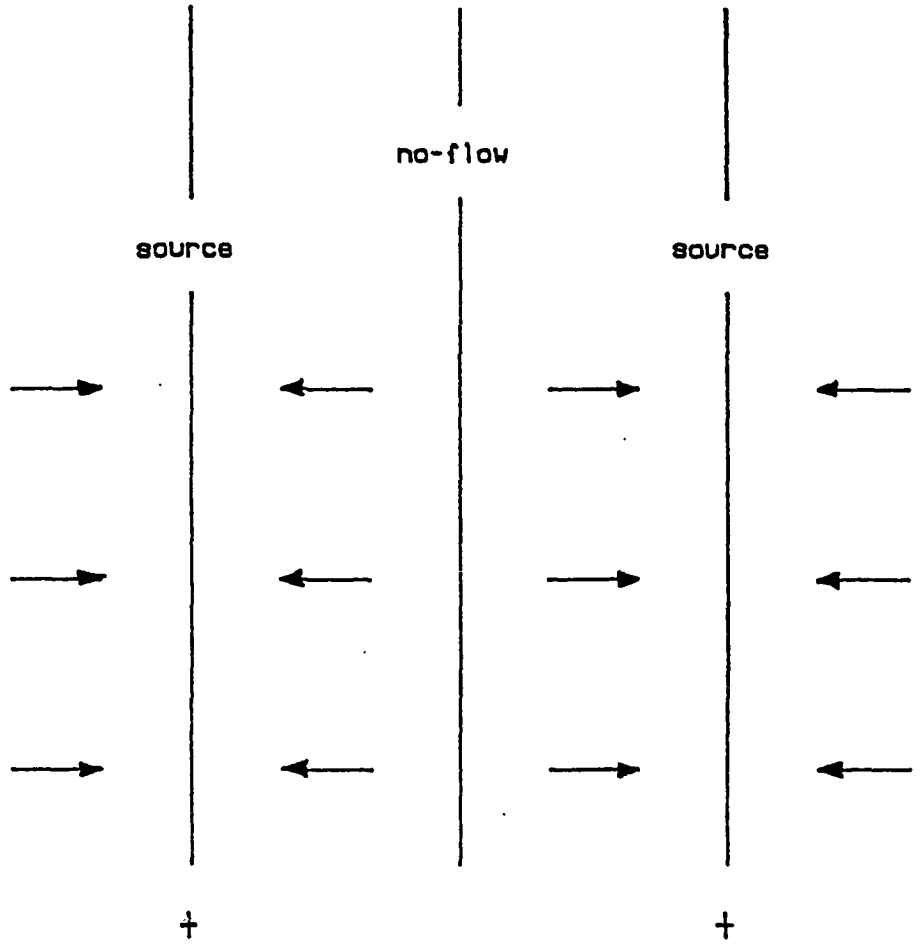


Fig. 2.4. Creation of a no-flow boundary.

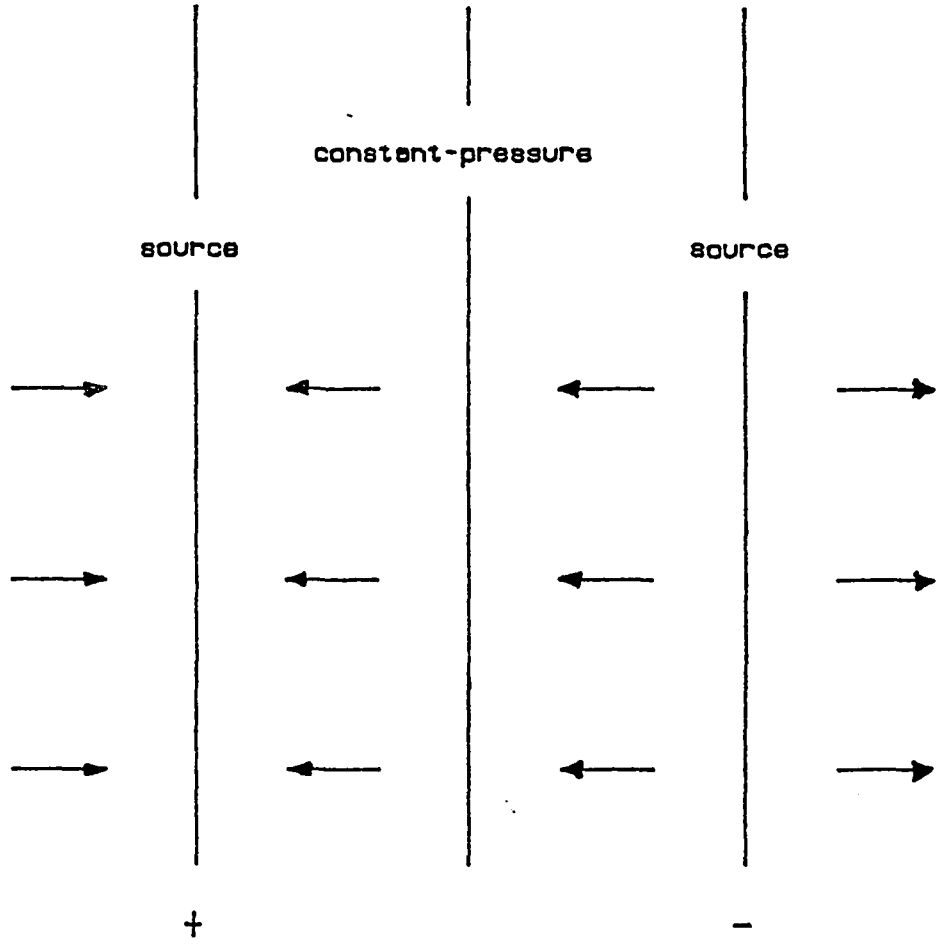


Fig. 2.5. Creation of a constant-pressure boundary.

an image source is subtracted from another source, a constant-pressure boundary results, as shown in Fig. 2.5. Again, the unit strength assumption places the boundary halfway between the sources. This boundary is constant-pressure because the pressure drop due to the source is exactly compensated for by the image source.

This process can be extended to produce more than one boundary; care must be taken in accounting for all of the images that two or more boundaries will create. Since only one-dimensional source functions are sought, the only relevant case of multiple boundaries is the case of two parallel boundaries with the source located somewhere between them. For example, two no-flow boundaries with a plane source within results in the set of images shown in Fig. 2.6.

The infinitude of images can be inferred by looking at how half of them are generated: The source must have an image to the left of A. It must be added to create the no-flow boundary. Because this image will induce flow across B, it must be compensated for by an image of equal distance from B to the right of B. This image then implies another image further to the left of A, and so on. These image sources must all be added in order to preserve the no-flow boundaries. The other half of the images are arrived at by using the same reasoning, except the image of the source through B is considered instead. This logic can be used to generate the case of one no-flow combined with one constant-pressure boundary, and for the case of two constant-pressure boundaries. The only difference is that the source functions of the images (and of the images of the images) will be

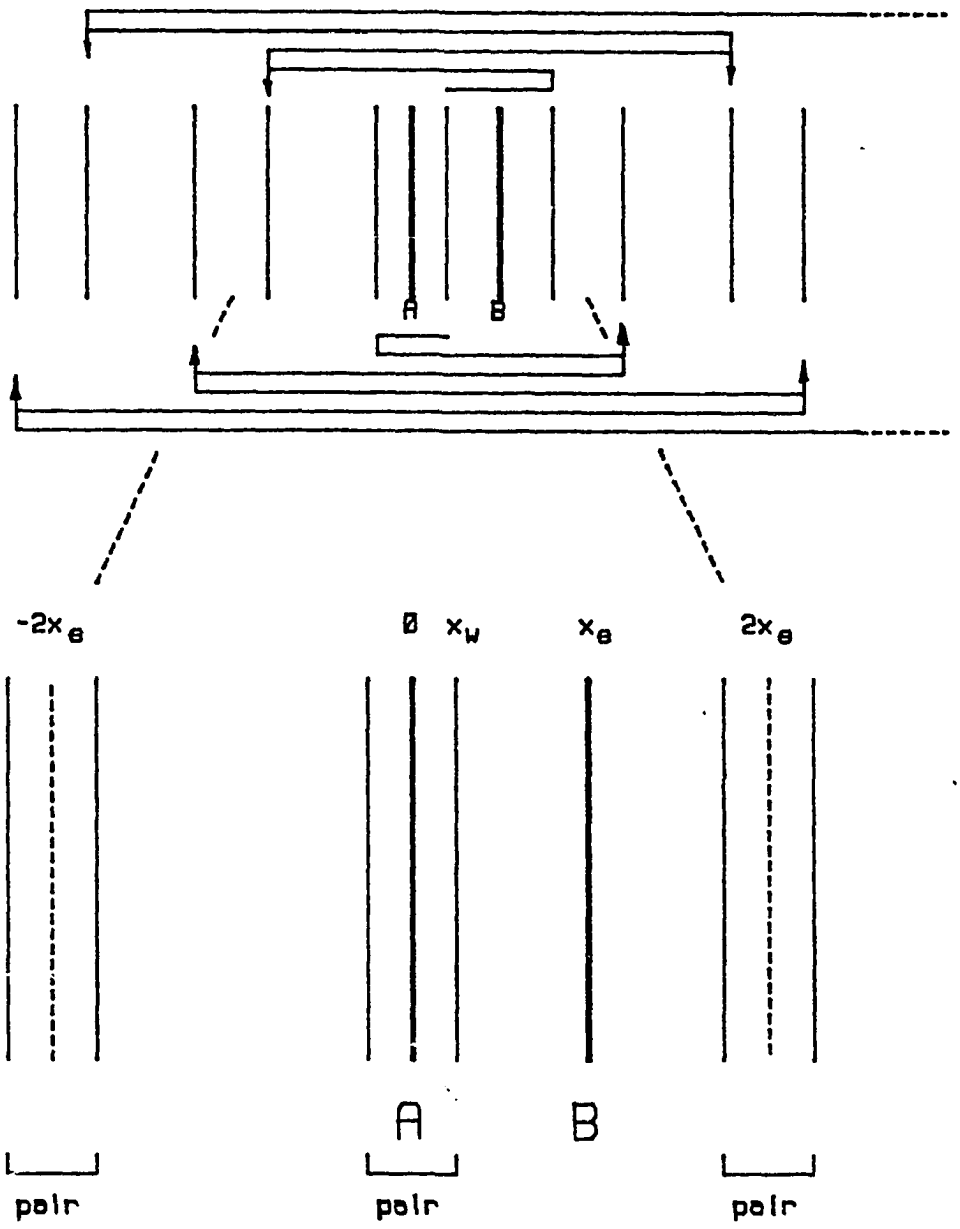


Fig. 2.6. Images and source pairing created by two boundaries.

added or subtracted, depending on the boundary which that particular image was reflected through. The results for the three boundary-pair combinations described, and for both the plane source and the slab source are derived as follows.

Using the same nomenclature and geometry as for the Laplace forms, the no-flow + no-flow case can be constructed as in Fig. 2.7, with all sources being positive due to imaging through no-flow boundaries. Because sources conveniently group in pairs, each pair separated by the distance x_e, it is helpful to denote each pair by the index j, where the pair for j = 0 includes the source within the boundaries. Then, for each pair, the two distances to the observation point x are expresed as:

$$[x-(2jx_e + x_w)] \text{ and } [x-(2jx_e - x_w)] \tag{2.25}$$

For VII'(x) , we assume plane sources, so each source contribution is of the form of I(x). Thus:

$$VII'(x) = \frac{1}{2\sqrt{\pi\eta\tau}} \sum_{j=-\infty}^{+\infty} \left[\exp\left[\frac{-(x-(2jx_e+x_w))^2}{4\eta\tau}\right]\right.$$

$$\left. + \exp\left[\frac{-(x-(2jx_e-x_w))^2}{4\eta\tau}\right]\right] \tag{2.26}$$

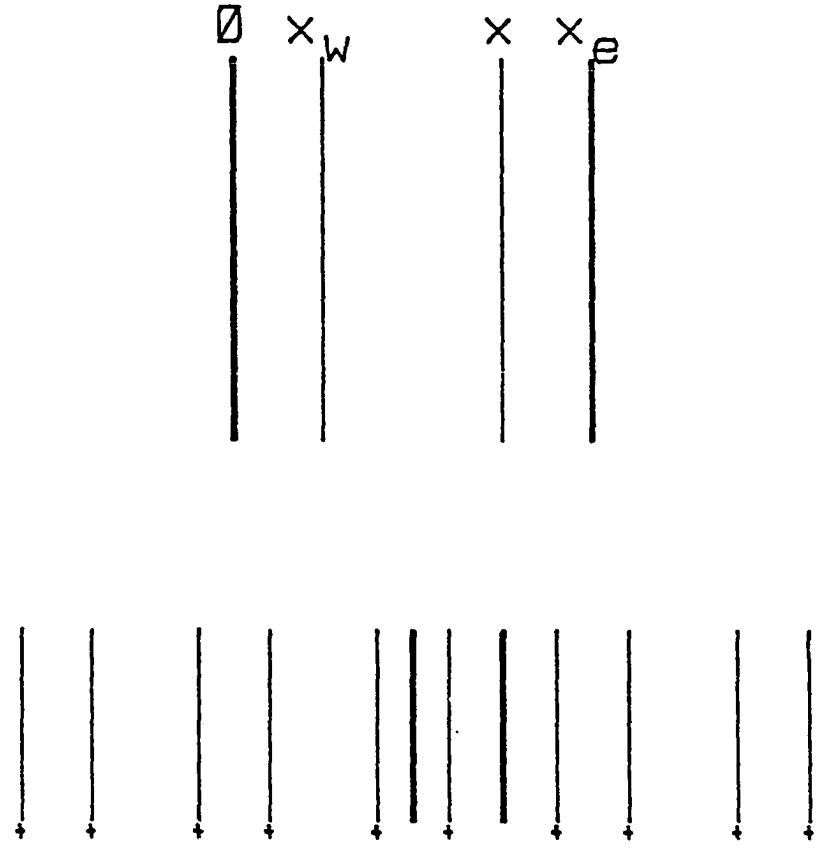


Fig. 2.7. Nomenclature and pair signs for no-flow + no-flow case.

In a similar fashion, the constant-pressure + constant-pressure case can be constructed (Fig. 2.8), except that, for each pair of sources, one is positive and one is negative. The two distances needed are the same as Eq. 2.25, but the left source of the pair is subtracted, so:

$$VIII'(x) = \frac{1}{2\sqrt{\pi\eta\tau}} \sum_{j=-\infty}^{+\infty} \left[\exp\left[\frac{-(x-(2jx_e+x_w))^2}{4\eta\tau}\right]\right.$$

$$\left. - \exp\left[\frac{-(x-(2jx_e-x_w))^2}{4\eta\tau}\right]\right] \qquad (2.27)$$

For the mixed boundary case, the situation becomes that of Fig. 2.9. Now each pair of sources has the same sign, but the sign alternates from pair to pair. Because the sign of the source function inside the bounds must be positive, the sign of the j^{th} pair behaves like $(-1)^j$, so that:

$$IX'(x) = \frac{1}{2\sqrt{\pi\eta\tau}} \sum_{j=-\infty}^{+\infty} (-1)^j \left[\exp\left[\frac{-(x-(2jx_e+x_w))^2}{4\eta\tau}\right]\right.$$

$$\left. + \exp\left[\frac{-(x-(2jx_e-x_w))^2}{4\eta\tau}\right]\right]\} \qquad (2.28)$$

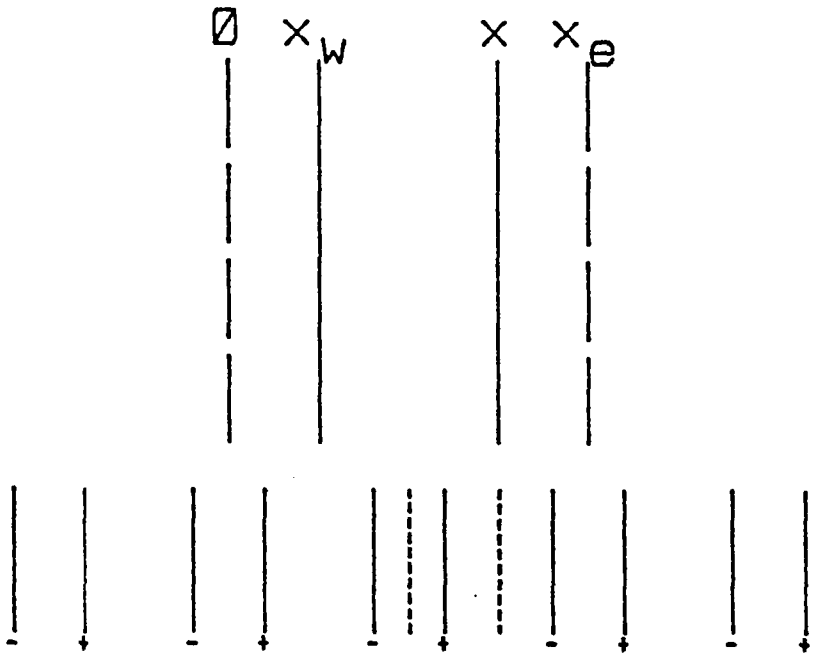


Fig. 2.8. Nomenclature and pair signs for const.-p. + const.-p. case.

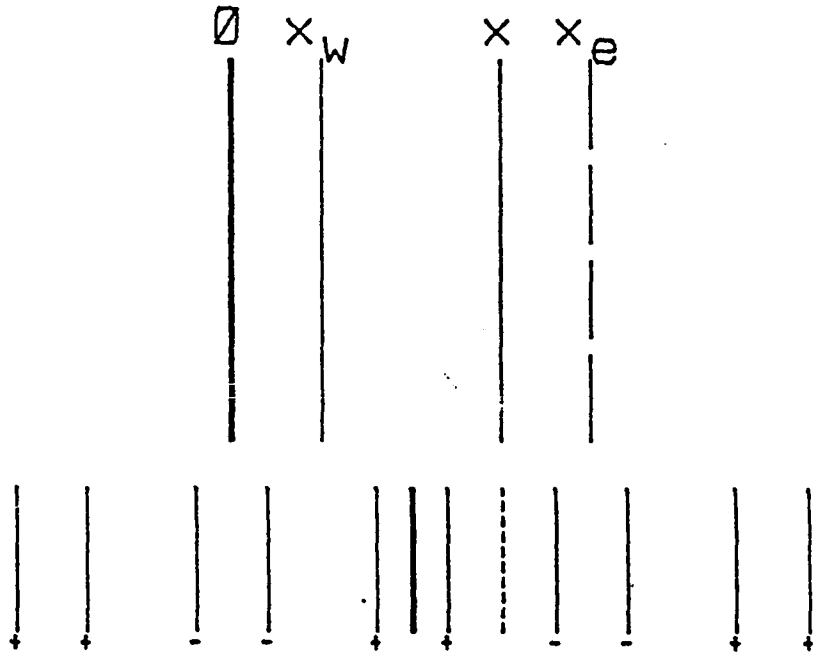


Fig. 2.9. Nomenclature and pair signs for mixed boundary case.

For the slab sources, the diagram, source pairing, distance functions and sign conventions are identical (with the exception of the source, which is a slab of thickness x_f centered at x_w), so the new source functions for these cases are presented without further discussion:

$$X'(x) = \frac{1}{2} \sum_{j=-\infty}^{+\infty} \left[\operatorname{erf} \left(\frac{x_f/2+(x-(2jx_e-x_w))}{2\sqrt{\eta\tau}} \right) + \operatorname{erf} \left(\frac{x_f/2-(x-(2jx_e-x_w))}{2\sqrt{\eta\tau}} \right) \right.$$

$$\left. + \operatorname{erf} \left(\frac{x_f/2+(x-(2jx_e-x_w))}{2\sqrt{\eta\tau}} \right) + \operatorname{erf} \left(\frac{x_f/2-(x-(2jx_e-x_w))}{2\sqrt{\eta\tau}} \right) \right] \tag{2.29}$$

$$XI'(x) = \frac{1}{2} \sum_{j=-\infty}^{+\infty} \left[\operatorname{erf} \left(\frac{x_f/2+(x-(2jx_e+x_w))}{2\sqrt{\eta\tau}} \right) + \operatorname{erf} \left(\frac{x_f/2-(x-(2jx_e+x_w))}{2\sqrt{\eta\tau}} \right) \right.$$

$$\left. - \operatorname{erf} \left(\frac{x_f/2+(x-(2jx_e x_w))}{2\sqrt{\eta\tau}} \right) - \operatorname{erf} \left(\frac{x_f/2-(x-(2jx_e-x_w))}{2\sqrt{\eta\tau}} \right) \right] \tag{2.30}$$

$$XII'(x) = \frac{1}{2}\sum_{j=-\infty}^{+\infty}(-1)^j\left[\operatorname{erf}\left(\frac{x_f/2+(x-(2jx_e+x_w))}{2\sqrt{\eta\tau}}\right) + \operatorname{erf}\left(\frac{x_f/2-(x-(2jx_e+x_w))}{2\sqrt{\eta\tau}}\right)\right.$$

$$\left. + \operatorname{erf}\left(\frac{x_{f/2}+(x-(2jx_e-x_w))}{2\sqrt{\eta\tau}}\right) + \operatorname{erf}\left(\frac{x_{f/2}-(x-(2jx_e-x_w))}{2\sqrt{\eta\tau}}\right)\right] \tag{2.31}$$

The prime notation has been chosen to signify that these new, alternate forms are identical in function to their non-prime counterparts, and thus can be substituted for the old forms wherever appropriate.

It can be seen that, by taking the limit of VII'(x) as time approaches zero, the limiting behavior is that of just a single plane source (one half of the pair, being beyond the boundary, does not contribute), the intuitive conclusion desired for that case because the boundaries do not affect flow at early times. Similarly, VIII'(x) and IX'(x) have this same limit, and X'(x) through XII'(x) have the single slab source as their limit. Since the source functions must be integrable from zero to the desired time, as per Eq. 2.6, it is very useful to have source

functions which can be evaluated at small times. Conversely, the limits of equations VII(x) through XII(x) as time approaches zero are not the plane or slab source forms of I(x) or II(x) at all, and, in fact, the limits do not exist. Computationally, when the series is evaluated for small times approaching zero, the number of terms needed becomes so large as to make their evaluation impracticable. The new forms avoid this problem completely. These new forms cannot be practically used by themselves either, because the number of terms needed to produce an answer for late times becomes as cumbersome as for the Laplace forms at early times. It is reasonable to assume that the pressure drop may be needed for very late times as well as for early times. Reservoir limit testing precludes this fact. By integrating the alternate forms (VII'(x) through XII'(x)) for the early times and switching to the Laplace forms at some suitable switching point, accurate values can be obtained numerically in the shortest possible time (considering the method used). Clearly, a method of using the alternate forms in concert with their Laplace form counterparts is desirable from a computational point of view, and will be presented in the next chapter.

PROCEDURE

With the source functions defined by Gringarten and Ramey (1973), and the new alternate forms defined previously for use when the time is small, the object is to find a computational link between them, and to exploit this characteristic so that the source functions can be calculated efficiently for any desired geometry. Presented in this chapter are the steps used to make the various source functions compatible with each other. A method for assembling them in any desired fashion first requires that all solutions generated have identical definitions for all the dimensionless variables, rather than the definitions being different for different geometries. Hence, a discussion of the proper selection of dimensionless variables which provide the necessary generality and flexibility is presented first. Then, the method used in this work for taking advantage of the alternate source functions for early-time calculations is illustrated. Finally, a discussion of how one can put the above results to immediate use is presented.

A. Dimensionless Variables.

The use of dimensionless variables in Green's function solutions has been demonstrated by many authors, including those cited in the Introduction . In each case, the solution for the desired geometry was specified first, then a set of dimensionless variables was chosen, and the solution was restated in dimensionless form. This has

resulted, in some cases, in arbitrary and highly specific definitions for the dimensionless variables, making comparisons of the various published solutions difficult. It will be shown that, in order to preserve generality, dimensionless variables can be chosen without knowing in advance the form of the solution. The conflicting definitions, which will result (for the time variable only), can be easily resolved afterwords by noting certain characteristics of their interactions.

Dimensionless time is defined as:

$$t_D = \frac{kt}{\phi\mu c[\ell_t^2]} = \frac{\eta t}{[\ell_t^2]} \qquad (3.1)$$

where $[\ell_t^2]$ is dimensionally equal to squared length and can be any convenient length defined for the system. Dimensionless pressure is defined as:

$$p_D = \frac{2\pi k[\ell_p]\Delta p}{q\mu} \qquad (3.2)$$

where Δp is provided by the integration of the chosen Green's functions, q is the flow rate for constant rate production, and $[\ell_p]$

is again any convenient length defined for the system. Thus, from Eq. 2.19,

$$\Delta p = \frac{q\mu p_D}{2\pi k[\ell_p]} = \frac{1}{\phi c V_s} \int_0^t q(\tau)\, S\,(x, t-\tau)\, d\tau \qquad (3.3)$$

Using Eq. 3.1,

$$t = \frac{\phi\mu c[\ell_t^2]}{k}\, t_D \qquad (3.4)$$

$$dt = \frac{\phi\mu c[\ell_t^2]}{k}\, dt_D \qquad (3.5)$$

Thus:

$$\frac{q\mu p_D}{2\pi k[\ell_p]} = \frac{q\phi\mu c[\ell_t^2]}{\phi c k V_s} \int_0^{t_D} S_D(\bar{x}_D, \tau_D)\, d\tau_D \qquad (3.6)$$

where S_D is a new source function which depends on dimensionless time and position instead of actual time and position, and $q(\tau) = q$ for constant rate production. By cancellation:

$$P_D = \frac{2\pi[\ell_p][\ell_t^2]}{V_s} \int_0^{t_D} S_D(\bar{x}_D, \tau_D)\, d\tau_D \qquad (3.7)$$

If S_D can be derived, then P_D vs. t_D can be calculated, regardless of the value of the outer reservoir parameters. Since S and S_D are the product of two or three source functions, each member of the product S_D must be expressible in terms of the same dimensionless time. Some suitable choice of $[\ell_t^2]$ needs to be made in order to achieve this. Further, the choice must not force any of the source functions to be associated with any particular dimension, or else the flexibility of choice for each axis is lost.

The method proposed in this work is to define the most convenient dimensionless time for any given source function, and to resolve the differences between the times outside of the function. This is most easily accomplished by introducing dimensionless lengths, i.e. ratios of two different lengths. For example, consider VII(x) again. If the lengths were combined into ratios, it would become:

$$VII_D(x) = \frac{1}{x_e}\left[1+2\sum_{n=1}^{\infty} \exp(-n^2\pi^2\tau_D)\cos n\pi x_{wD}\cos n\pi x_D\right]$$

(3.8)

where:

$$x_{wD} = \frac{x_w}{x_e} \tag{3.9}$$

$$x_D = \frac{x}{x_e} \tag{3.10}$$

$$\tau_D = \frac{\eta t}{x_e^2} \tag{3.11}$$

and $[\ell_t^2] = x_e^2$. If this is done for all of the source functions, with dimensionless lengths and times chosen as advantageously as possible for each, dimensionless source functions result, which are listed in Table 3.1. Source functions I(x), II(x), and VI(x) have deliberately been excluded from this table. The boundaries in VII'(x) and X'(x) can be placed arbitrarily far away from the source so that, for all practical purposes, the boundaries do not affect the pressure response, thus the functions are equivalent to I(x) and II(x), respectively. Source function VI can be simulated by three instances of VII'(x) together, again with boundaries made arbitrarily far away.

It is important to note that some source functions, such as III and IV, or VII through IX, have factors in front of

them, and the rest do not. This inverse length factor will be denoted $1/[\ell_s]$ so that $[\ell_s]$ has the dimension of length or squared length. Taking this into account along with Eq. 3.7 gives:

$$p_D = \frac{2\pi[\ell_p][\ell_t^2]}{[\ell_s]V_s} \int_0^{t_D} S_D(\bar{x}_D, \tau_D)\, d\tau_D \qquad (3.12)$$

For any combination for source functions, the term:

$$\frac{[\ell_p][\ell_t^2]}{[\ell_s]V_s} \qquad (3.13)$$

must be resolved before p_D can be calculated.

As it turns out, V_s and $[\ell_s]$ are intimately related; the product of these two terms is always dimensionally a cubed length, regardless of the choices of source functions. For example, the line source combined with the slab source as in Example 2.1, $V_s = z_e$, since the length of the line is the only dimensionality of the source. From Table 3.1, the first term of the equation gives $[\ell_s]$ as per the definition given in the text above. This is summarized in the following equations:

$$[\ell_s] = [x_e][y_e][1] = x_e y_e \qquad (3.14)$$

$$V_s = [1][1][z_e] = z_e \qquad (3.15)$$

$$[\ell_s]V_s = x_e y_e z_e \qquad (3.16)$$

$[\ell_s]$, V_s , and $[\ell_s]V_s$ for each source function can be summarized as in Table 3.2. Note that, for the radial sources, $[\ell_s]V_s$ is dimensionally a squared length, and for each of the linear sources, the product is just a length. The final value of $[\ell_s]V_s$ used is again the product of the $[\ell_s]V_s$ for each source function, just as the final source function integrated is the product of the individual source functions. Thus, for a radial source and any linear source, the final value of $[\ell_s]V_s$ is a cubed length, and for any three linear sources together, that final value is again a cubed length. Because $[\ell_s][\ell_t^2]$ is always a cubed length, the value of the constant of Eq. 3.13 is always dimensionless, as it must be.

The only problem remaining after making the source functions dimensionless is to convert the fixed definition of dimensionless time (Eq. 3.1) into the individual times defined for each source function. For example, if VII(x) is used in the z-direction of a cartesian, three-dimensional geometry (creating VII(z), with all

Table 3.2 $[\ell_s]$ and V_s for the Dimensionless Source Functions.

source function	$[\ell_s]$	V_s	$[\ell_s]V_s$
$III_D(x_D)$	r_w^2	1	r_w^2
$IV_D(x_D)$	r_w	$2\pi r_w$	$2\pi r_w^2$
$V_D(x_D)$	1	$\pi(r_2^2-r_1^2)$	$\pi(r_2^2-r_1^2)$ $\{ = \pi r_w^2$ if $r_1 = o$ and $r_2 = r_w\}$
$VII_D(x_D)$, $VII_D'(x_D)$	x_e	1	x_e
$VIII_D(x_D)$, $VII_D'(x_D)$	x_e	1	x_e
$IX_D(x_D)$, $IX_D'(x_D)$	x_e	1	x_e
$X_D(x_D)$, $X_D'(x_D)$	1	x_f	x_f
$XI_D(x_D)$, $XI_D'(x_D)$	1	x_f	x_f
$XII_D(x_D)$, $XII_D'(x_D)$	1	x_f	x_f

variables converted to the z-axis), the dimensionless time which the source function needs for calculation is $\eta t/z_e^2$. This implies that conversions are needed between $[\ell_t^2]$ and x_e^2, y_e^2, and z_e^2. If $[\ell_t^2]$ is chosen appropriately, these conversions are simplified. The above discussion applies in concept to the radial, two-dimensional problem as well. The best values to choose for $[\ell_p]$ and $[\ell_t^2]$, therefore, depend on the choice of geometry, and are given in the next section.

B. Separation of Cylindrical and Rectangular Geometries.

For both conceptual and practical reasons, the cylindrical problems were separated from the rectangular ones and developed separately. The practical reasons include not only the obvious desire to limit the scope of any programming project to avoid unnecessary complications, but also because optimal choices for $[\ell_t^2]$ and the length factor of Eq. 3.13 are related to the geometry. For the cylindrical cases,

$$[\ell_t^2] = r_w^2 \qquad (3.17)$$

$$[\ell_p] = z_e \qquad (3.18)$$

With these assumed, the possible forms of Eq. 3.13 are provided in Table 3.3. The results show that the factor $1/\pi$ or $1/2\pi$ can be chosen when the corresponding cylindrical function is chosen, and a $1/z_{fD}$ factor can be included if a slab source version of a z-axis source function is chosen. Dimensionless time conversion is not needed for the radial source function, and thus the relationship between z_e and r_w is the only parameter needed. In this work, the dimensionless variable z_e/r_w, which represents the thickness of the formation in wellbore radii, is used, and the time conversion is simply the square of the inverse of this quantity.

For the rectangular problems, the results are just as simple; when it is assumed that:

$$[\ell_t^2] = x_e y_e \tag{3.19}$$

$$[\ell_p] = z_e \tag{3.20}$$

another tabulation of the possible forms of Eq. 3.13 can be generated. This is shown in Table 3.4. Again, a simple rule of thumb can be inferred from the results, namely that for every slab source form of source function used, a factor of $1/i_{fD}$ is generated, where i stands for the dimension of that slab function.

Table 3.3 Constant Preceding Pressure Integral for Radial Forms

radial source	z-source	$\frac{[\ell_p][\ell_t^2]}{[\ell_s]V_s}$
III	VII, VIII, IX	1
III	X, XI, XII	$1/z_{fD}$
IV	VII, VIII, IX	$1/2\pi$
IV	X, XI, XII	$1/2\pi\ z_{fD}$
V	VII, VIII, IX	$1/\pi$
V	X, XI, XII	$1/\pi\ z_{fD}$

Table 3.4 Constant Preceding Pressure Integral for Linear Forms

x-source	y-source	z-source	$\frac{[\ell_p][\ell_t^2]}{[\ell_s]v_s}$
VII, VIII, IX	VII, VIII, IX	VII, VIII, IX	1
"	"	X, XI, XII	$1/z_{fD}$
"	X, XI, XII	VII, VIII, IX	$1/y_{fD}$
"	"	X, XI, XII	$1/y_{fD}z_{fD}$
X, XI, XII	VII, VIII, IX	VII, VIII, IX	$1/x_{fD}$
"	"	X, XI, XII	$1/x_{fD}z_{fD}$
"	X, XI, XII	VII, VIII, IX	$1/x_{fD}y_{fD}$
"	"	X, XI, XII	$1/x_{fD}y_{fD}z_{fD}$

The conversion of the dimensionless time is facilitated by defining two dimensionless variables relating the relative magnitudes of the sizes of the outer boundaries. By defining $R_{yD} = y_e/x_e$ and $R_{zD} = z_e/x_e$, the dimensionless time implied by Eqs. 3.1 and 3.19 together can be converted to the time needed by the x-axis source function by multiplying by R_{yD}. The conversion for the y-axis source function time is $1/R_{yD}$, and for the z-axis, R_{yD}/R_{zD}^2. The definition of these two ratios is natural, since some relationship must be specified between the outer boundaries, or else the shape of the parallelepiped will not be known.

C. Crossover Time.

In the computational sense, there is a time when both the Laplace forms of Gringarten and Ramey (1973) and the new alternate forms of the previous chapter require the same amount of time to arrive at an answer. Before this time, the alternate form will always be faster, and after this time, the Laplace form will be faster. Since it is difficult to determine how many terms in the infinite series are needed to assure convergence on an answer, a crossover device was designed for the purpose of switching between the two forms while attempting to minimize computation time. To demonstrate this difficulty, source function $VII_D(x)$ will be examined. Recalling that $VII_D(x)$ is:

$$\frac{1}{x_e}\left[1+2\sum_{n=1}^{\infty}\exp\left(-n^2\pi^2\tau_D\right)\cos n\pi x_{wD}\cos n\pi x_D\right] \qquad (3.21)$$

It can be seen that the magnitude of each term can be no greater than the exponential part, and is probably less, since the sine terms will, in general, be out of phase with each other, and thus their product will have a magnitude less than or equal to one. In order to calculate how many terms are needed, a definition of convergence must be proposed. Generally, if an accuracy of one part in 10^6 is desired, then it can be said that when the ratio of the n^{th} term in a decreasing series to the sum of all the previous terms is less than 10^{-6}, that computation has finished. This is especially true of alternating series, because later terms tend to cancel rather than accumulate. The sine terms make the series of $VII_D(x)$ an alternating series. The magnitude of the n^{th} term must be less than or equal to:

$$\exp\left(-n^2\pi^2\tau_D\right) \qquad (3.22)$$

However, there is no simple relation for the magnitude of the sum of the previous terms since the behavior of the sine terms together is sensitive to phase, via the values of the variables in their

arguments. The sum of (n-1) exponential terms cannot be used as an estimator because it will be invariably too high. Without a good estimation of the magnitude of the previous terms, no reliable estimate of the number of terms needed for a given accuracy can be made. The implication here is that it is desired to use $VII'_D(x)$ for a certain time, until it takes more time than $VII_D(x)$ to calculate, then switch over to VII(x) permanently; the number of terms needed for each must be determined during a series of computations, and then a choice made at some point to switch over. In other words, the behavior of the series over a time span cannot be inferred in advance, so some computational device must be used to monitor the progress of the source functions, and also to choose which to use. A decision-making aid was developed to address this problem.

Each source function is monitored for the number of terms it takes to reach the specified tolerance. If, after a computation of a value for the source function for a specific value of time, the number of terms required for the computation exceeded a preset number (bound), a flag that controls which source function to use is set to true. At the same time, its bound on term count is increased by a moderate amount. When another calculation is desired, the other form of the source function will be used which counts the number of terms it needs, and checks its own bound. If the term count is exceeded again, its upper bound is increased and the flag is set to false. This can best be described graphically, as in Fig. 3.1.

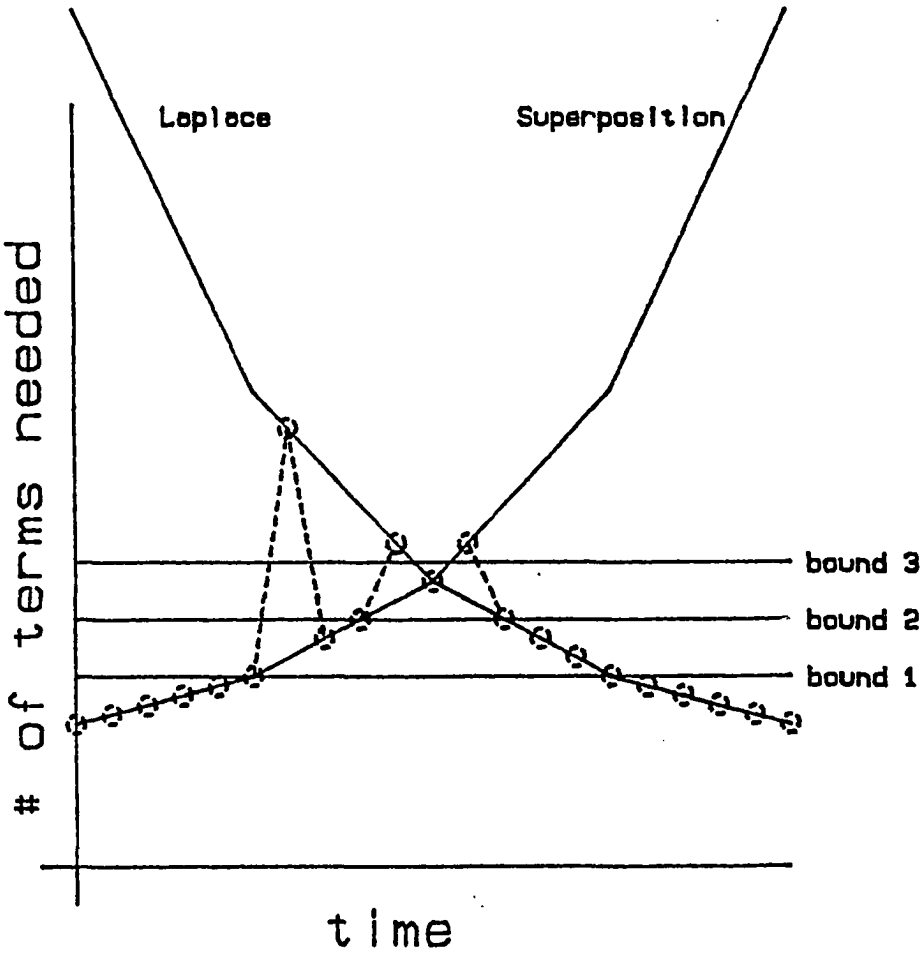


Fig. 3.1. Crossover time, term bounds, and source function chosen.

Integration of the source functions is performed for exponentially-increasing time. The early-time form is used by default for the earliest specified time. As long as its term count bound is not exceeded, this form continues to be used. At some time, the bound will be too low and the Laplace form will be used next. The early form's bound is increased. If the Laplace form term count bound is broken, the choice is switched back and the Laplace form term bound is increased. Eventually the term bounds will be high enough to make the Laplace form the function of choice. Since the number of terms the Laplace form will need will continue to decrease, it will be used until the time has reached a desired upper limit.

D. User Implementation.

The process for constructing a pressure solution is very simple when the tables of the previous sections are used. First, the geometry must be chosen. Then, the diagrams of Table 2.1 should be consulted in order to decide which source functions, when multiplied together, give both the desired well geometry and the desired boundary type. The source functions to use are listed in dimensionless form in Table 3.1. This table is consulted, and the chosen source functions are extracted. As noted before, certain source functions are not included in Table 3.1 because of their superflous nature. The unbounded cases for linear sources can be easily simulated by using the bounded cases and later setting the values of the dimensionless

variables so as to make the practical limits as far from the well as desired.

The dimensionless source functions from Table 3.1 are multiplied together, to form the source function S_D mentioned earlier. By consulting Table 3.3 for the radial cases, and Table 3.4 for the linear cases, the value of the constant factor of Eq. 3.13 can be found. Finally, the answer is given by Eq. 3.12, with the constant factor and S_D (determined above) substituted as shown.

To complete the discussion of Example 2.1, the dimensionless solution will be derived here using the method outlined above. The appropriate source functions to use are again VII(x), VII(y), and X(z) (unity), as discussed previously. The dimensionless forms of these functions are given in Table 3.1. The early-time form is considered first. Using $VII'_D(x)$ and $VII'_D(y)$ ($X'_D(z)$ also unity) as given in the table, the resulting S_D for this case is:

$$[VII'_D(x,R_{yD}\tau_D)] \cdot [VII'_D(y,\tau_D/R_{yD})] \qquad (3.23)$$

By consulting Table 3.4, the constant factor for the solution can be found, as is $1/z_{fD}$. Therefore, using Eq. 3.12, the pressure solution is:

$$p_D = \frac{2\pi}{z_{fD}} \int_0^{t_D} [VII_D'(x,R_{yD}\tau_D)] \cdot [VII_D'(y,\tau_D/R_{yD})]\, d\tau_D \qquad (3.24)$$

The time factors used in each source function have been resolved by using the conversion given in the preceding section.

Using the same exact method, the Laplace forms can be combined to give the solution as well. The S_D for this case is:

$$[VII_D(x,R_{yD}\tau_D)] \cdot [VII_D(y,\tau_D/R_{yD})] \qquad (3.25)$$

and, using the same constant factor as before, the pressure solution is:

$$p_D = \frac{2\pi}{z_{fD}} \int_0^{t_D} [VII_D(x,R_{yD}\tau_D)] \cdot [VII_D(y,\tau_D/R_{yD})]\, d\tau_D \qquad (3.26)$$

By careful examination, using the definitions of the dimensionless variables given earlier, Eq. 3.26 and Eq. 2.23 will be found to be identical.

Because of the high degree of regularity in the above approach, it is an easy matter to translate the process given into computer terms, where the functions can be assembled and the times converted without user intervention. Two programs, one for the radial cases and one for the rectangular cases, are given in the Appendix. The user of either of these programs merely specifies the geometry, any conversions desired between the default dimensionless time and pressure definitions to the desired time and pressure definitions (such as basing time on fracture length, as is convenient for fracture studies), and the values of the dimensionless variables applicable to the geometry. Practical application to two recently published cases is provided in the next chapter.

APPLICATIONS

The concepts of the preceding chapters are applied to two recent problems described in the literature: (a) the parallelepiped model for geothermal steam reservoirs, and (b) the elongated linear flow system. The purpose of this chapter is to illustrate practical uses of the methods that have been developed in this work.

A. Fractured Geothermal Reservoir.

Cinco, et al. (1979) used the source function approach to model a geothermal steam reservoir which was naturally fractured and which was hypothesized to contain a horizontal boiling interface some distance below the geothermal well completion interval. The reservoir was believed to be sealed by faults around its sides, and on the top by impermeable cap rock. They presumed that the boiling interface was equivalent to a constant-pressure surface, since an isothermal reservoir would maintain the boiling interface at the vapor pressure. Drawdown tests of the Travale 22 well in the Larderello Field in Italy showed characteristic half-slope behavior on log-log plots of pressure versus time, indicating the presence of fractures, followed by a trend towards a stable, constant-flowing pressure.

The above conditions were readily modelled by the source functions, with no-flow boundaries in the x- and y-directions, and mixed boundaries in the z-direction. The fracture (oriented in the x-z plane) was modelled by two slab sources coupled with one plane

source. Using the notation of the previous chapter, the source functions utilized were X(x), VII(y), and XII(z). A diagram of this configuration is given in Fig. 4.1. The nomenclature used for this case was typical for fracture studies. In particular, all dimensions in the geometry were made dimensionless by dividing them by the fracture half-length, which the authors symbolized by x_f. This definition of x_f is contrary to the definition implied by the source functions used to define the geometry. In other words, their definition is different from the original definition given by Gringarten and Ramey (1973), and likewise is different from that used throughout this work, in which x_f is defined as the full fracture length. To further complicate matters, h_f was given as the fracture height, while z_f was given as the vertical position of the center of the fracture, and the fracture length and fracture height do not both use the concept of half-length. This usage of two coordinate names for the z-axis is not helpful, and the inconsistent use of the fracture half-length is problematic.

In this work, the definitions of fracture height z_f and fracture position z_w are consistent with the definitions of fracture length and fracture position that are used in the x-direction. The formation thickness is z_e, again from the source function definitions, not h, which is the traditional designation, but which is again inconsistent with the nomenclature used for the other two dimensions of the reservoir. Lastly, the reservoir is x_e by y_e areally, not $2x_e$ by $2y_e$ as used by Cinco, et al.

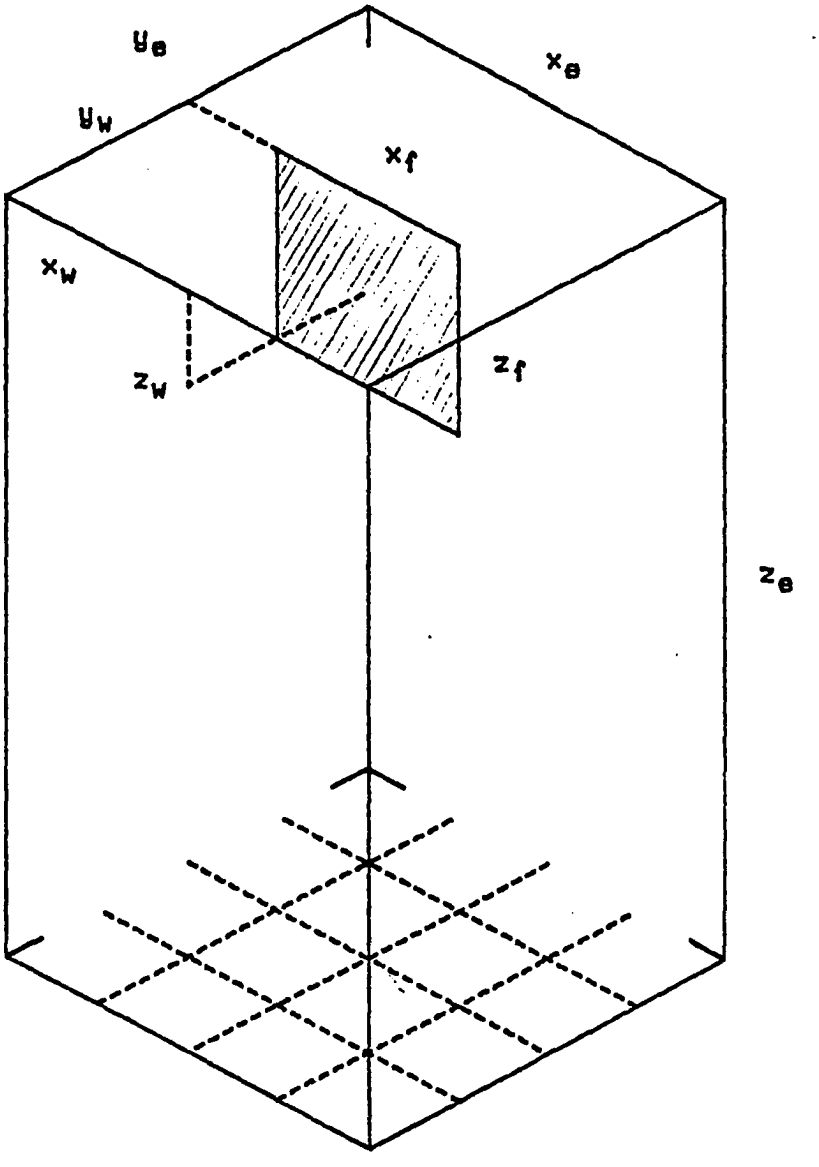


Fig. 4.1. Fractured geothermal well and reservoir.

In order to verify the published results, a transformation was derived to convert variables from the confusing published notation to the more consistent notation proposed here. This transformation is given in Table 4.1. As the transformation shows, changing some of the variables implies changes in more than one of the reservoir parameters simultaneously. A concise illustration of this problem is realized if one holds every dimension of the reservoir model fixed, except for the fracture half-length x_f. In that case, all of the dimensionless variables of Cinco, et al. would simultaneously change inversely, while use of the variables proposed in this work allows only one variable to change linearly, namely x_{fD}. In general, when pressure response varies in some characteristic way to a simple variable such as z_{fD}, this characteristic response can be easily isolated from responses for other changes imposed. By changing only one dimension of the geometry at a time, the sensitivity of pressure to that one factor is displayed.

As an extension to the analysis presented by Cinco, et al., the effect of fracture location within the reservoir is illustrated. The published solutions assume that the fracture is a 2:1 rectangle (half as high as it is wide), and that it is located in the center of a square ($x_e = y_e$) with depth twice that of the width or length of the drainage volume. Cases which vary the fracture shape or the reservoir shape could easily be addressed by the program, but were not attempted for this work. The fracture can move within the reservoir in two characteristic ways - either edge-on or face-on. The results for

edge-on movement are shown in Fig. 4.2, and for face-on movement in Fig. 4.3. Edge-on movement disturbs the development of pseudo-radial flow, which is confirmed by the gradual loss of the pseudo-radial portion of the curve. An artificial linear section develops, but this effect is coincidental, and not indicative of any particular flow regime. Therefore, the lack of a pseudo-radial section in some pressure data might indicate that the fracture edge was near a no-flow boundary. The effect of fracture movement is even more marked for the face-on case. Not ony is pseudo-radial behavior modified, but pseudo-steady-state-type behavior becomes a factor, as evidenced by one of the curves increasing in slope from the typical 1/2 for fractures before it flattens as steady-state is reached. This is due to restricted flow to the one side of the fracture located closer to the boundary. This result is qualitatively similar to that given by Gringarten and Ramey (1974) for the case of horizontal fractures and restricted entry. The most important conclusion to draw from these results is that if the fracture is not centered, the curvature due to pseudo-radial flow can disappear, making a unique match less likely.

Cinco, et al. provided data from a pressure test for a geothermal steam well believed to conform to the geometry studied. The type-curve match which they demonstrated shows good agreement with the pressure curve for $x_e/x_f = 2$ and $h_D = 10$ (their variables). By using the transform given above, and by specifying the match values for h_D, position, and so on, the method of this work reproduced the pressure curve and match given by Cinco, et al., and is shown in Fig. 4.4. The

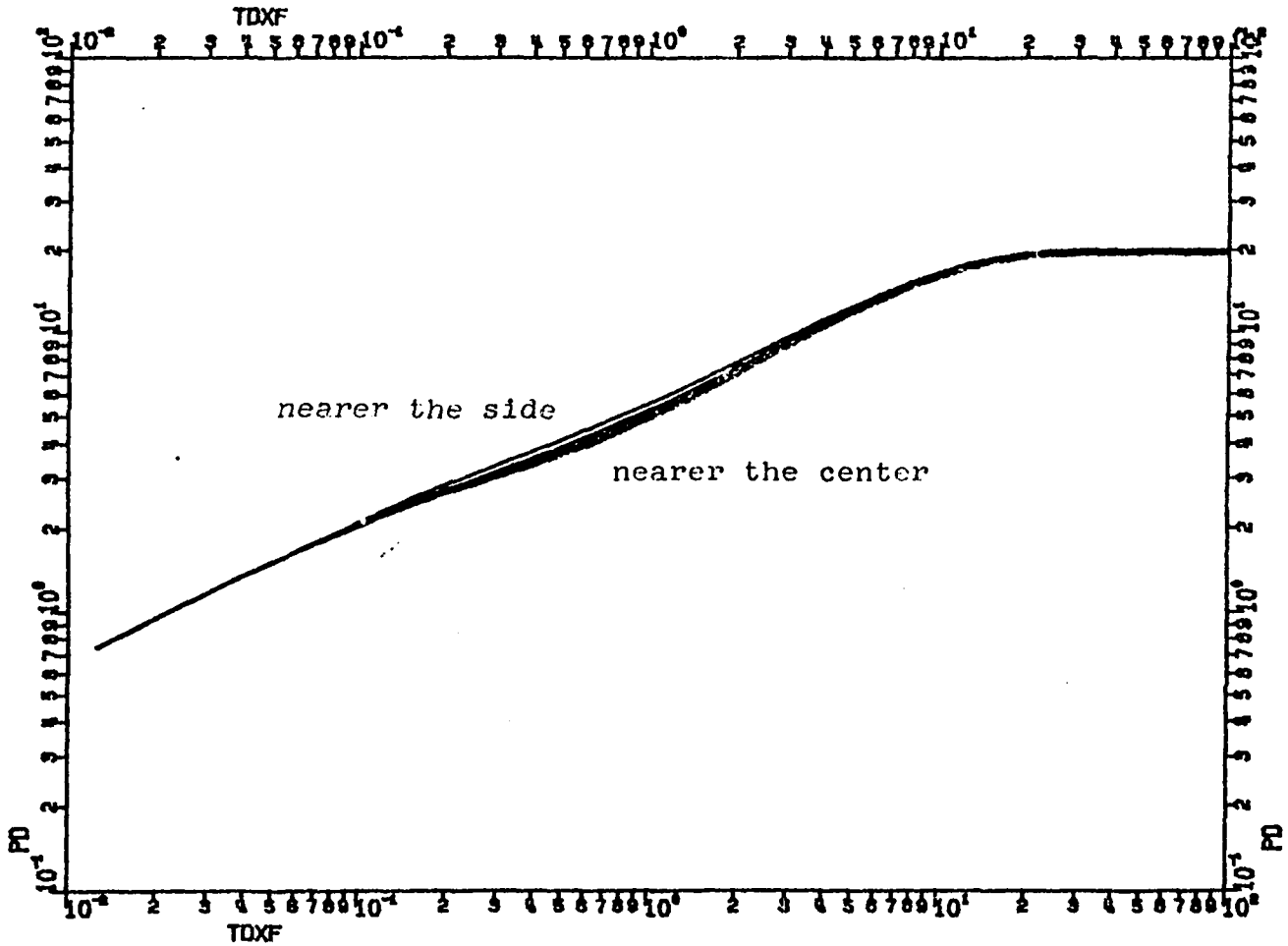


Fig. 4.2. The effect of edge-on movement of fracture.

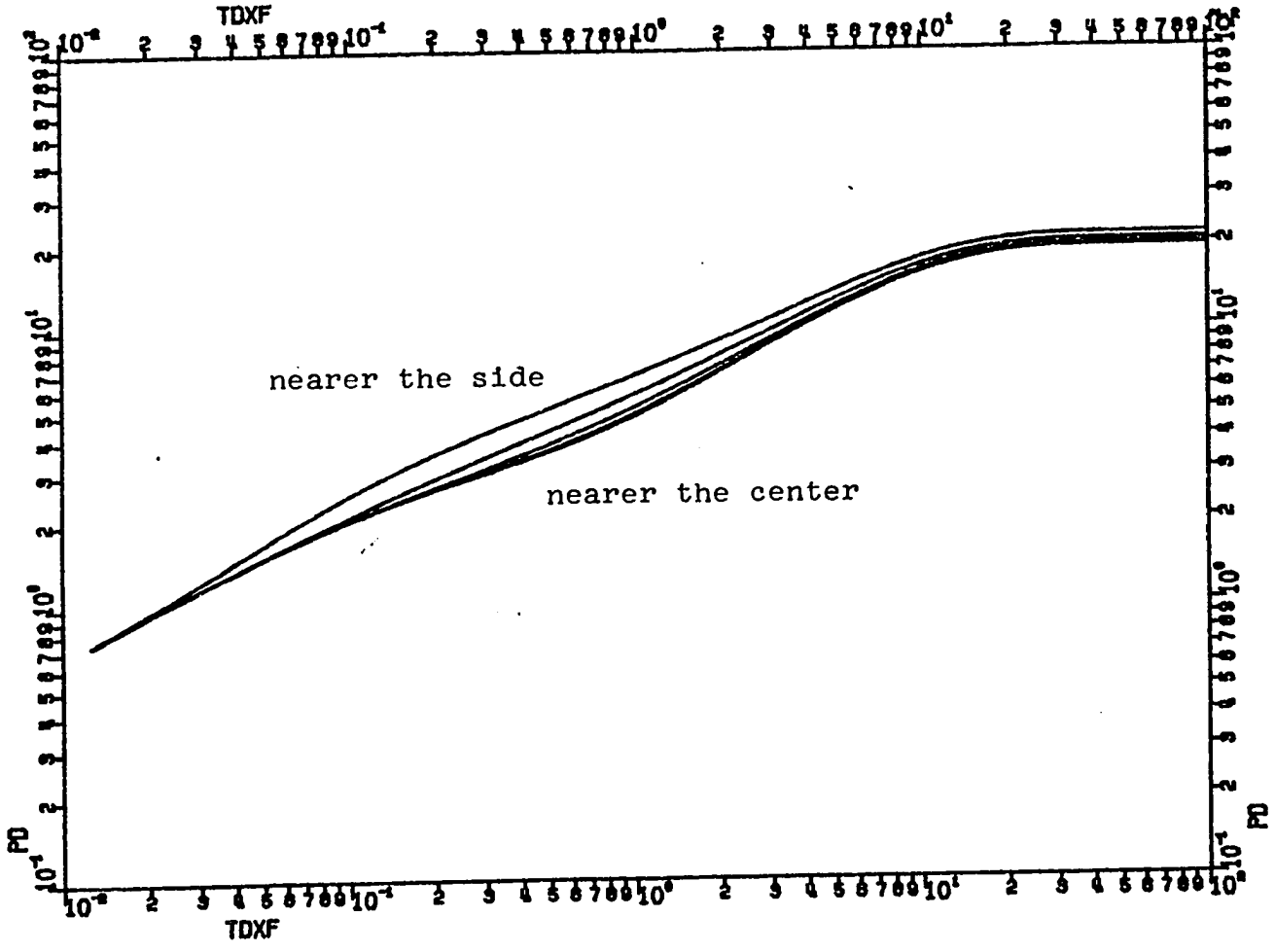


Fig. 4.3. The effect of face-on movement of fracture.

values of the dimensionless variables used to generate this curve are given in Table 4.2.

For geothermal steam wells, dimensionless pressure is defined somewhat differently from the conventional definition, which assumes volumetric flow of a liquid and pressure drop in form $(p_i - p_{wf})$. Steam wells are treated in the same fashion as natural gas wells, where mass flow of gas is used, and the pressure drop is measured as $(p_i^2 - p_{wf}^2)$. Dimensionless pressure is defined as:

$$p_D = [.01442] \frac{Mkz_e \Delta p^2}{ZTq\mu} \qquad (4.1)$$

where:

- M = molecular weight of steam (18 lb/lb-mole)
- k = permeability (darcys)
- z_e = steam zone thickness (meters)
- Δp^2 = $(p_i^2 - p_{wf}^2)$ = change in squared pressure kg^2/cm^4
- Z = steam compressibility factor (.85 this test)
- T = reservoir temperature (553°K this test)
- q = mass flow rate of steam (205 ton/hour this test)
- μ = steam viscosity (.019 cp this test)

Dimensionless time is:

$$t_{Dxf} = [.3604] \frac{kt}{\phi \mu c x_f^2} \qquad (4.2)$$

where:

t = time (hours)

ϕ = porosity (.16 this test)

c = isothermal compressibility (.01 cm^2/kg this test)

x_f = fracture length (meters)

The analysis by Cinco, et al. of the type-curve match is given below. By choosing the point indicated on Fig. 4.4 as the match point, the following can be determined:

t = (80 days)*(24 hours/day) $\qquad t_{Dxf} = 5.2$

$\Delta_p^2 = 747\ kg^2/cm^4 \qquad p_D = 25$

Using Eq. 4.1 and the given test data,

$$(25) = (.01442) \frac{(18)[k \cdot z_e](747)}{(.85)(553)(205)(.019)} \qquad (4.3)$$

$$[k \cdot z_e] = 236.1 \quad \text{darcy-meter} \qquad (4.4)$$

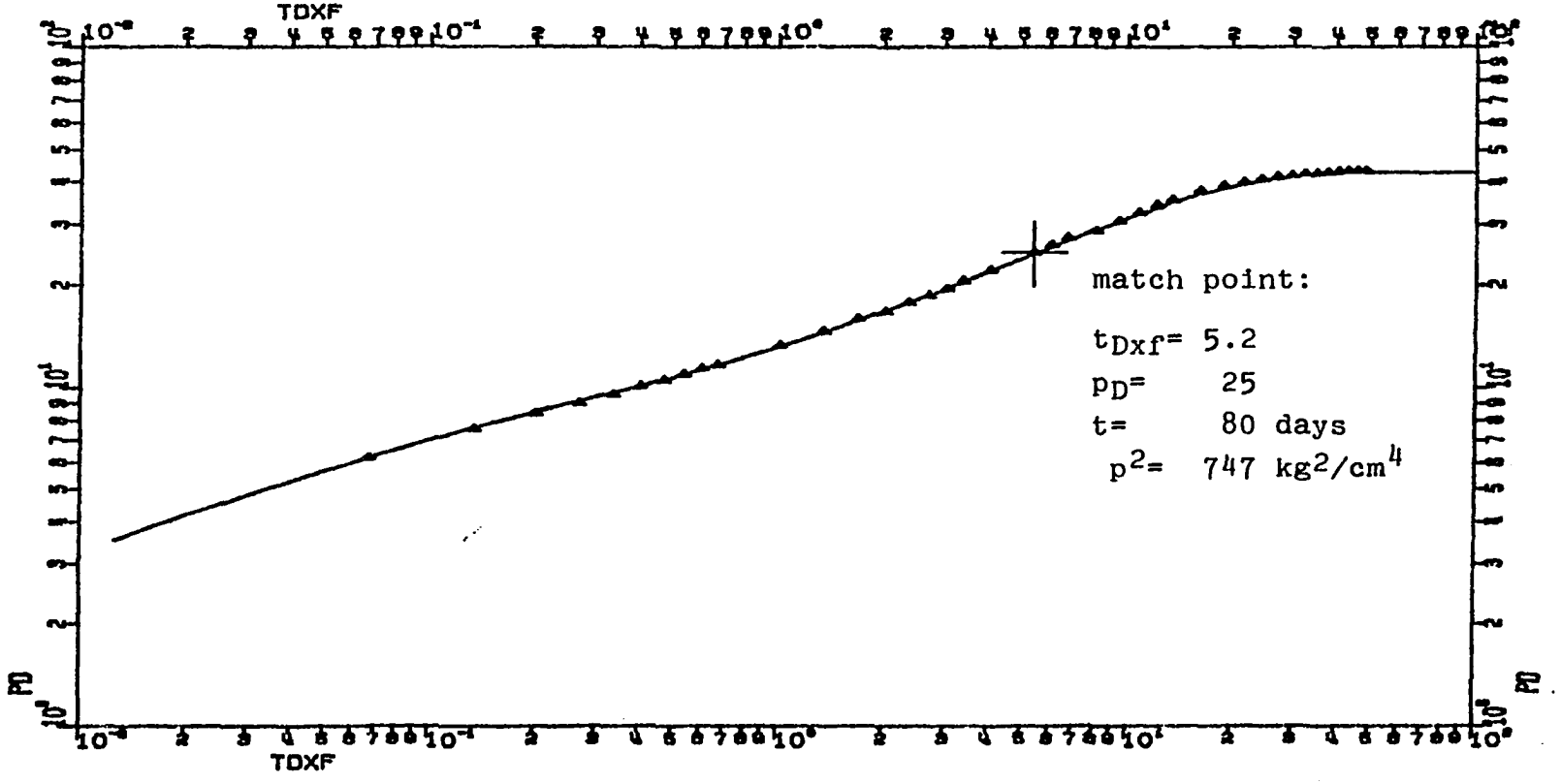


Fig. 4.4. Type-curve match of Cinco, et.al.

Using Eq. 4.2,

$$(5.2) = (.3604)\frac{[k](80)(24)}{(.16)(.019)(.01)[x_f^2]} \qquad (4.5)$$

$$[k/x_f^2] = 2.2845 \times 10^{-7} \text{ darcy/meter}^2 \qquad (4.6)$$

To isolate the value of the permeability, Eq. 4.4 and 4.6 can be used as follows:

$$[k \cdot z_e]^2 \cdot [k/x_f^2] = [k^3] \cdot [x_e^2/x_f^2] \qquad (4.7)$$

Since:

$$[z_e^2/x_f^2] = [z_e/x_f]^2 = (5)^2 \qquad (4.8)$$

(note: $z_e/\frac{1}{2} x_f = 10 = h_D$)

it follows that:

$$[k^3] = \frac{(236.1)^2(2.2845 \times 10^{-7})}{(25)} \tag{4.9}$$

$$k = .0799 \text{ darcy} \tag{4.10}$$

The depth to the boiling interface, z_e, is determined by:

$$z_e = \frac{[k \cdot z_e]}{k} = \frac{(236.1)}{(.0799)} \tag{4.11}$$

$$z_e = 2955 \text{ meters} \tag{4.12}$$

The full fracture length, x_f, is calculated in a similar fashion:

$$x_f = \left[\frac{k}{[k/x_f^2]}\right]^{1/2} = \left[\frac{(.0799)}{(2.2845 \times 10^{-7})}\right]^{1/2} \tag{4.13}$$

$$x_f = 591.4 \text{ meters} \tag{4.14}$$

These results are identical with those of Cinco, et al.

The match of Fig. 4.4 is quite close, but may not be the only match possible. The program of this work was used interactively to determine whether an alternative match was possible. An example of an intermediate step is shown in Fig. 4.5, and the match which is implied is shown in detail in Fig. 4.6. To determine the match, the data from the well test was displayed on a graphics terminal, along with the desired family of type-curves generated by the model. In this case, the family of curves conform to the specifications given in Table 4.2, except that z_D, z_{wD}, and z_{fD} were allowed to vary, as R_{zD} was varied. The well test data was then "moved" on the screen to the match position shown in Fig. 4.5. A value of R_{zD} of 3 provided the best match for the last portion of the data, so this particular curve was graphed along with the data, and is shown in Fig. 4.6. For this case, z_{fD}=.1133, R_{zD}=3, and x_{fD}=5. By using the same calculation procedure as before,

$$(2.0) = (.01442)\frac{(18)[k \cdot z_e](441)}{(.85)(553)(205)(.019)} \qquad (4.15)$$

$$[k \cdot z_e] = 31.99 \text{ darcy-meter} \qquad (4.16)$$

$$(2.0) = (.3604)\frac{[k](20)(24)}{(.16)(.019)(.01)[x_f^2]} \qquad (4.17)$$

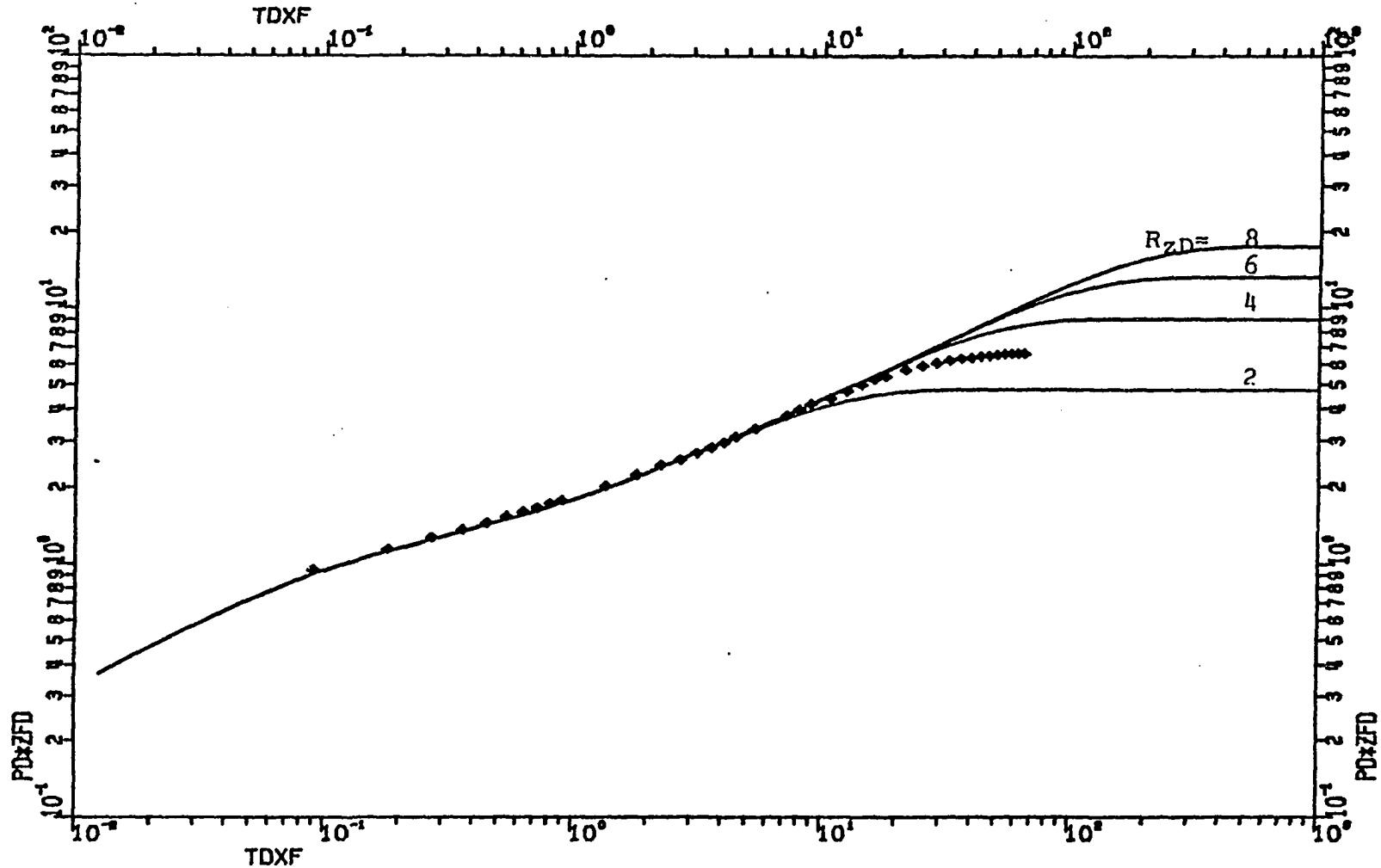


Fig. 4.5. Type-curve match within curve family.

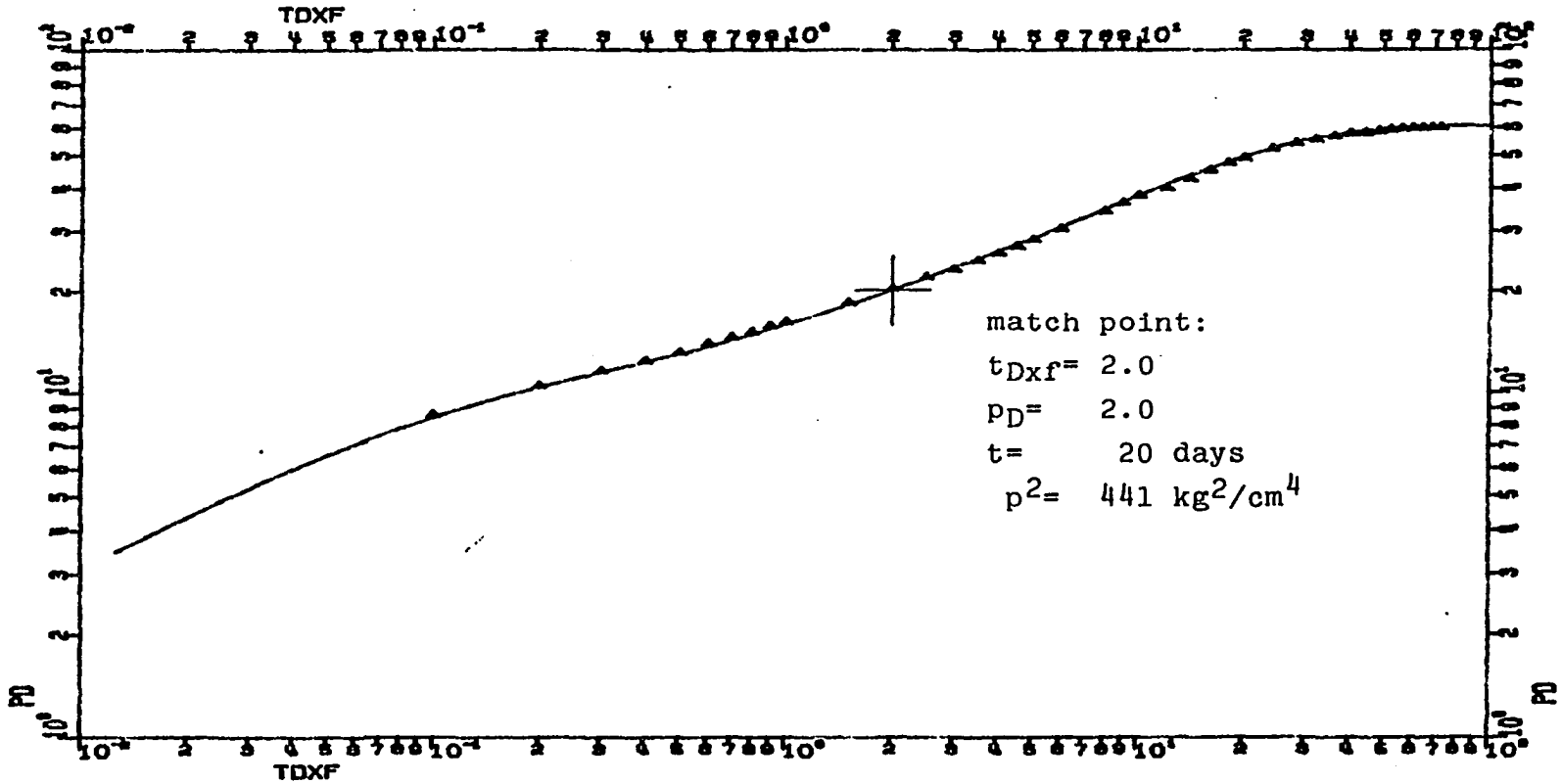


Fig. 4.6. Final type-curve match verification.

$$[k/x_f] = 3.5146 \times 10^{-7} \text{ darcy/meter}^2 \qquad (4.18)$$

$$[k \cdot z_e]^2 \cdot [k/x_f^2] = [k^3] \cdot [z_e^2/x_f^2] \qquad (4.19)$$

$$[z_e^2/x_f^2] = [z_c^2/x_e^2] \cdot [x_e^2/x_f^2] = [R_{zD}^2][1/x_{fD}^2] \qquad (4.20)$$

$$k = \left[\frac{31.99)^2(3.5246 \times 10^{-7})}{(3^2)(2^2)}\right] = .0215 \text{ darcy} \qquad (4.21)$$

$$z_e = \frac{(31.99)}{(.0215)} = 1488 \text{ meters} \qquad (4.22)$$

$$x_f = \left[\frac{(.0215)}{(3.5146 \times 10^{-7})}\right]^{1/2} = 247.3 \text{ meters} \qquad (4.23)$$

The differences in the calculated permeability values and depths to the boiling interface show that they are sensitive to how one chooses to match raw data to pressure curves. Table 4.2 gives a comparison of the values calculated for the two matches studied. The differences are quite significant, even though the geometries are very similar. The smaller depth of the second match, along with the smaller length and width, implies a much smaller steam zone. This

Table 4.1 Dimensionless Variable Transformation.

$x_D = .5 + .178/x_{De}$	$x_{wD} = .5$	$x_{fD} = 1/x_{De}$
$y_D = .5$	$y_{wD} = .5$	
$z_D = .268\ h_{fD}/h_D$	$z_{wD} = .5\ h_{fD}/h_{wD}$	$z_{fD} = h_{fD}/h_D$
$R_{yD} = 1$	$R_{zD} = h_D/x_{De}$	

Table 4.2 Comparison of Type-Curve Matches.

model	Cinco, et. al	alternate	Δ%
x_D	.683	.5	- 36.6
y_D	.5	.5	0.0
z_D	.0683	.0567	- 17.0
x_{wD}	.5	.5	0.0
y_{wD}	.5	.5	0.0
z_{wD}	.05	.0567	+ 13.4
x_{fD}	.5	.5	0.0
z_{fD}	.1	.1133	+ 13.3
R_{yD}	1.	1.	0.0
R_{zD}	2.5	3.	+ 20.0
K (darcys)	.0799	.0215	- 73.1
z_e (meters)	591.4	247.3	- 58.2
x_f (meters)	2455.	1488.	- 49.6
x_e (meters)	1182.8	494.7	- 58.2

might severely affect reserve estimation, and thus the value of the reservoir. It must be emphasized, though, that one or both of these matches may be incorrect for the given situation. The possibilities for configurations have not nearly been exhausted, so there may exist many other pressure matches, and thus many other values may be calculated, which may be near or far from those values already given.

This non-uniqueness of matching is typical in pressure transient analysis. The generality provided by the program created for this work allows the user who is performing the analysis a great deal of freedom in specifying the reservoir geometry. Those parameters which are known for the reservoir being studied can be fixed, and the remaining parameters can be varied one at a time until it is clear whether a unique geometry is implied by the data, or whether there are many possible interpretations.

B. The Linear Flow Channel.

Linear flow geometry is known to exist for water influx problems, and for the predominantly faulted reservoirs of many geothermal regions. Certain depositional environments are responsible for elongated reservoirs, such as river channels or meanders. More recently, linear flow has been deduced for many low permeability gas fields.

Nutakki and Mattar (1982) used superposition to develop solutions for a well centered in a flow channel of infinite length. Kohlhaas,

et. al. (1982) showed similar derivations, along with practical examples of pressure data which demonstrated the channel flow behavior they sought. The case of flow in a fractured channel has not been studied previously, and will be briefly discussed here. The curves to be presented are not meant to be used as type-curves in the conventional sense, although they could be. The purpose is to show how a computer, programmed to use the source function method of this work, can be used to make generalizations about pressure responses for a significant example.

The flow regimes which can be established during a pressure test in a fractured flow channel are diagrammed in Fig. 4.7. The first, fracture linear, represents the predominantly linear flow towards both sides of the fracture face. This effect is marked by a 1/2-slope straight line when log of pressure is plotted versus log of time. Pseudo-radial flow, so named because of its similarity to true radial flow, follows fracture linear flow, as long as no boundaries interfere. If boundaries disturb a portion, but not all, of the radial flow, it remains essentially radial for that portion of flow not interfered with, and the effect is similar. A plot of pressure versus log of time will show a straight line if radial or pseudo-radial effects occur. Channel linear flow represents linear flow along the channel distant from the fracture. This will occur after fracture linear flow, but may not necessarily occur after pseudo-radial flow. A plot of log of pressure versus log of time will show a 1/2-slope straight line, just as in the fracture linear case.

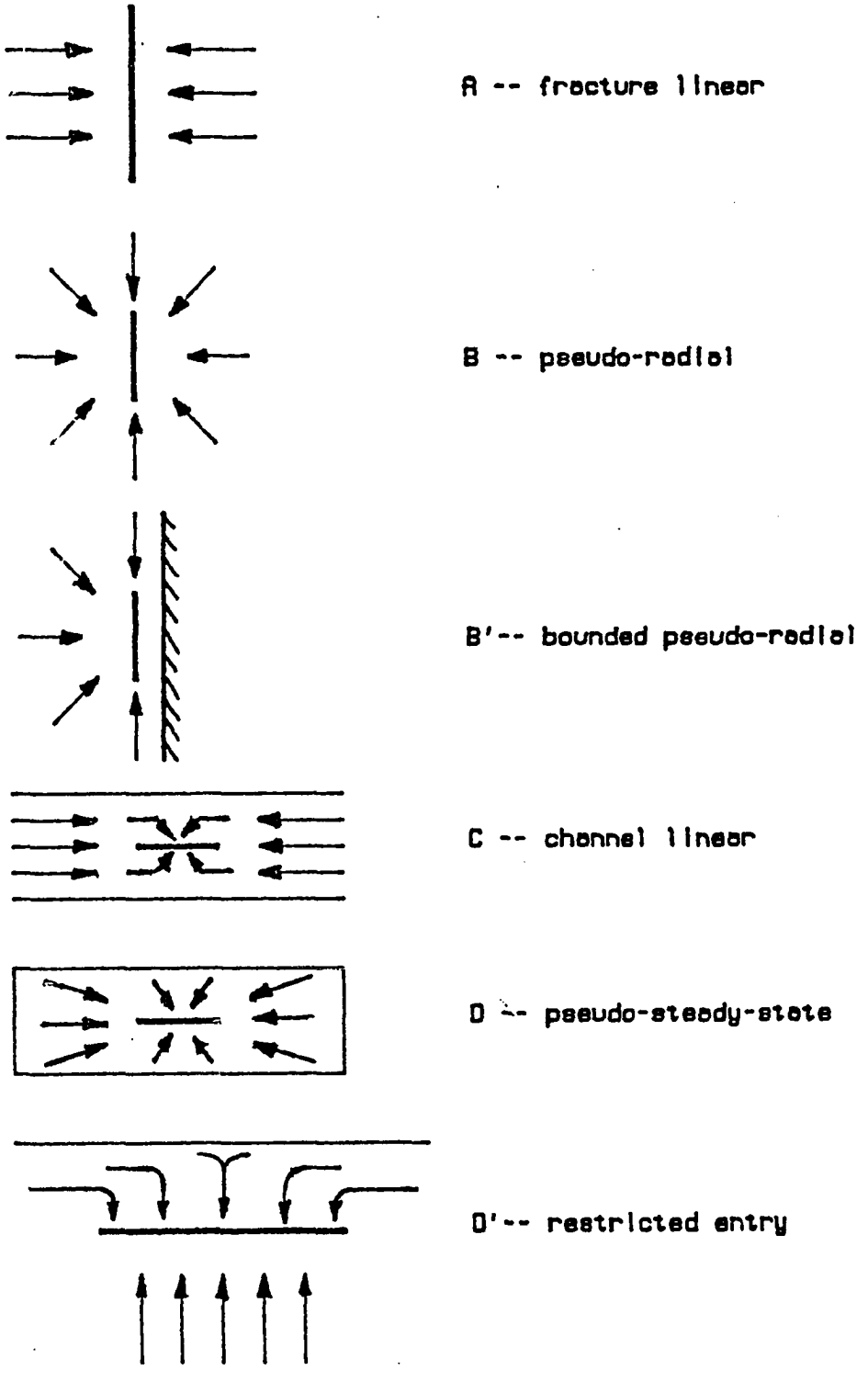


Fig. 4.7. Flow regimes, as seen from above.

Finally, since a real channel must have some finite length, pseudo-steady-state flow occurs. In this case, the pressure profile in the reservoir is constant, and the pressure drop at the well is due to the production of fluid only. A plot of log of pressure versus log of time will show a unit-slope straight line.

For simplicity, the pressure curves to follow will be labelled with the letters given in Fig. 4.7 at the appropriate points. The three configurations to be examined are shown in Fig. 4.8. The first will illustrate the formation of channel flow. The second will illustrate the effect of fracture position in the channel. The third will show the effect of fracture length as it relates to the channel width. Table 4.3 provides the data used to generate pressure curve families for these three cases. It must be emphasized at this point that two major assumptions were made: first, the channel is as deep as it is wide, and second, the fracture fully penetrates vertically. A number of variables are fixed by these assumptions; a complete examination of all the possibilities is beyond the scope of this work.

Fig. 4.9 shows the effect of lengthening the channel with everything else held constant. The initial 1/2-slope line is due to the fracture. For this case, the fracture length was accidentally chosen in such a way that pseudo-radial flow occured for only a short time, until the effects of the boundaries converted the flow pattern to that of channel linear. The onset of pseudo-steady-state flow becomes later as the channel length increases. The 1/2-slope line indicating channel flow is obviously the limit of this process as the

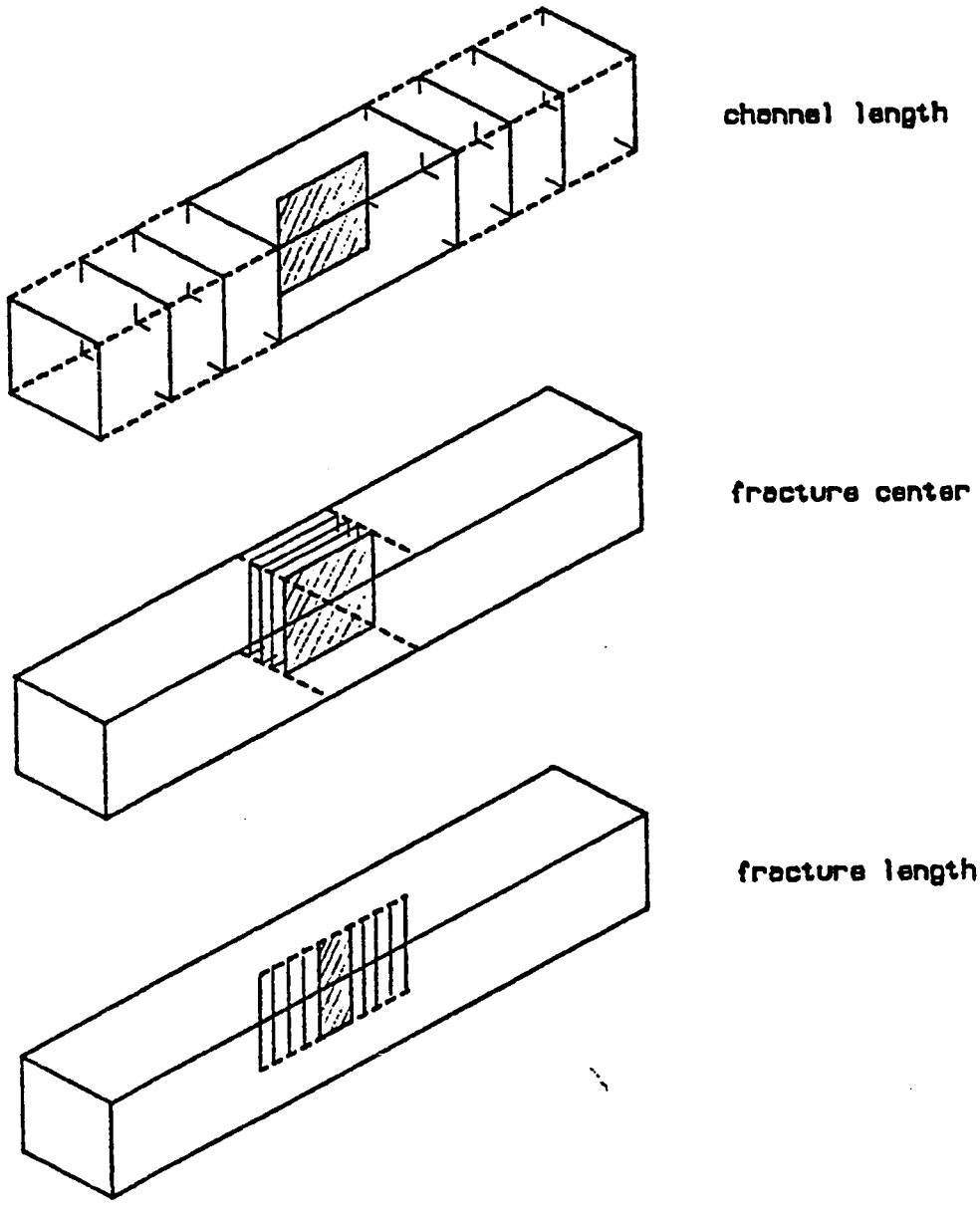


Fig. 4.8. Channel flow case studies (figures not to scale).

channel becomes infinitely long. For simple channel flow, then, there are three flow regimes: fracture linear, pseudo-radial (depending on fracture length; the last case will clarify this), and channel linear.

Fig. 4.10 demonstrates the effect of restricted entry to the fracture, as the fracture approaches a channel wall. Following the fracture linear flow, the normal pseudo-radial section is masked by the apparent pseudo-steady-state behavior caused by the near boundary. However, this restricted-entry behavior changes back to pseudo-radial, and finally to channel linear. The pressure response for $t_{Dxf} > 10^2$ is identical for all cases.

The final example illustrates the relationship between pressure response and fracture length as related to channel width. By specifying a channel length sufficiently long to prevent pseudo-steady-state from occuring, the channel width is the only controlling factor in the pressure deviations. The results are shown in Fig. 4.11. For fractures which are shorter than the channel is wide, pseudo-radial effects occur for a time inversely related to the fracture length, i.e. the shorter the fracture, the longer the pseudo-radial section will be before channel linear flow is established. For fractures which are longer than the channel length, effectively pseudo-steady-state behavior indicates more and more restricted entry into the fracture. The results are also qualitatively similar to those of moving the fracture near to a channel wall, except that there is no pseudo-radial flow at all. As was mentioned in the previous section, this is qualitatively similar to the behavior demonstrated by

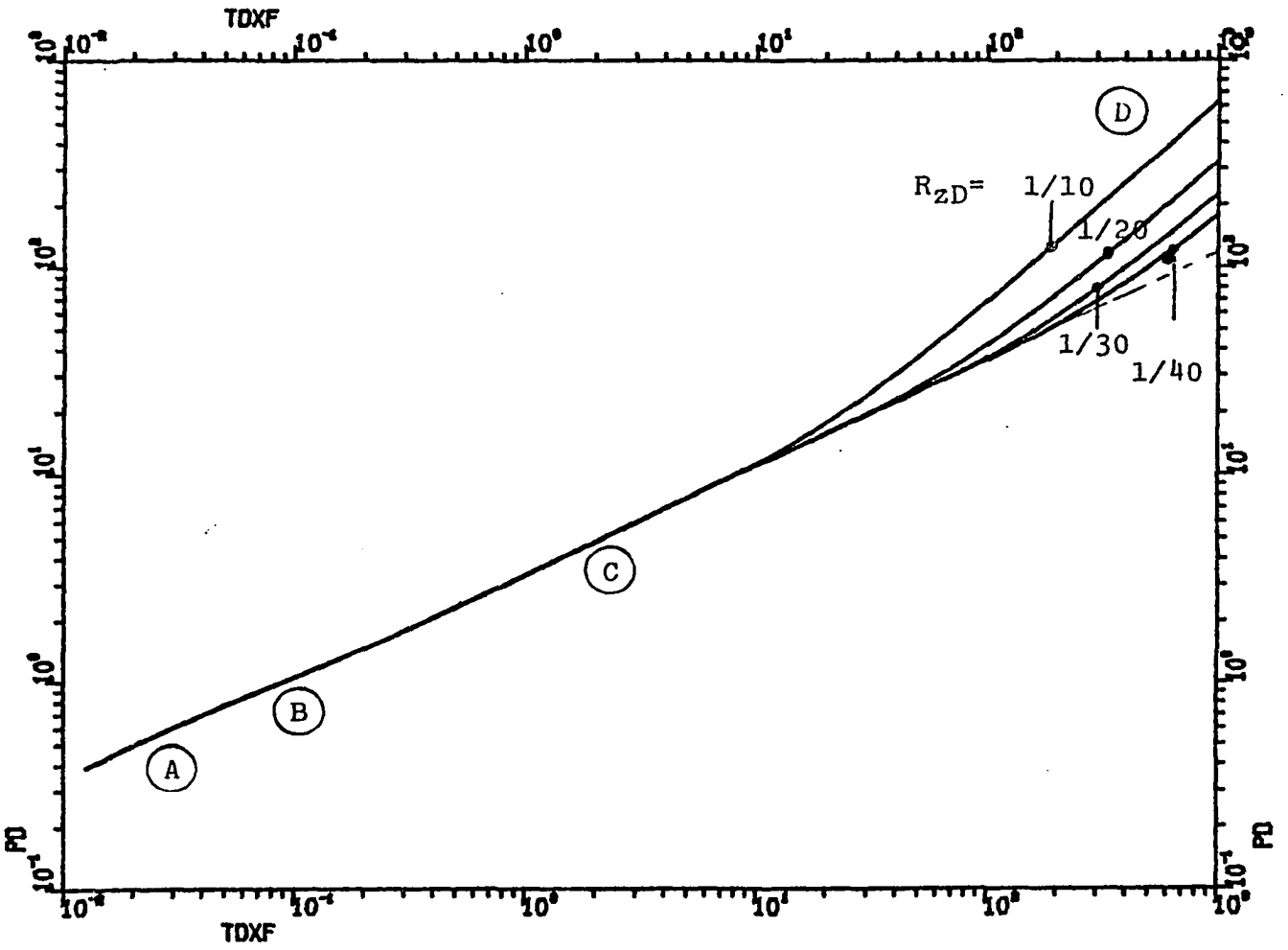


Fig. 4.9. Effect of lengthening of channel.

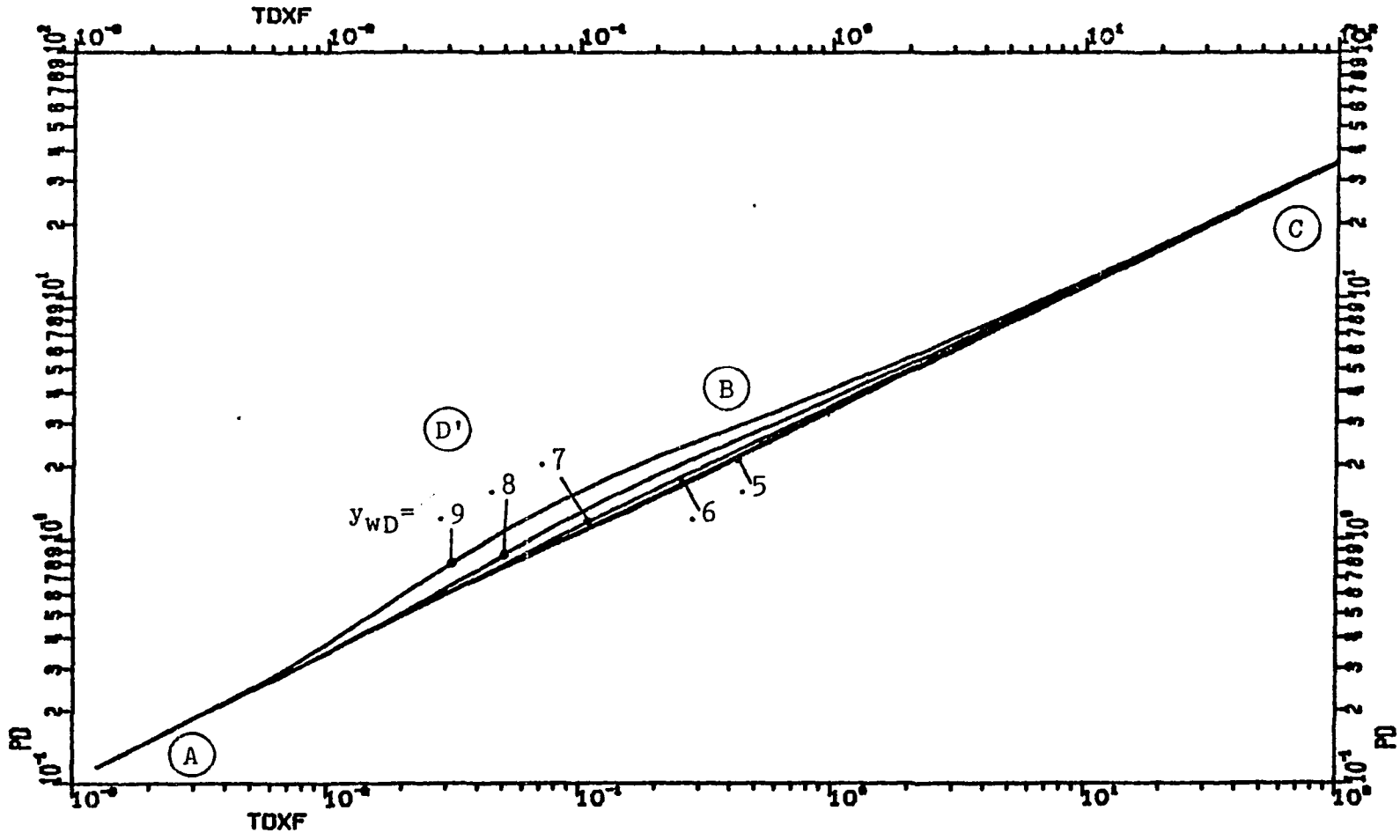


Fig. 4.10. Effect of fracture position within channel.

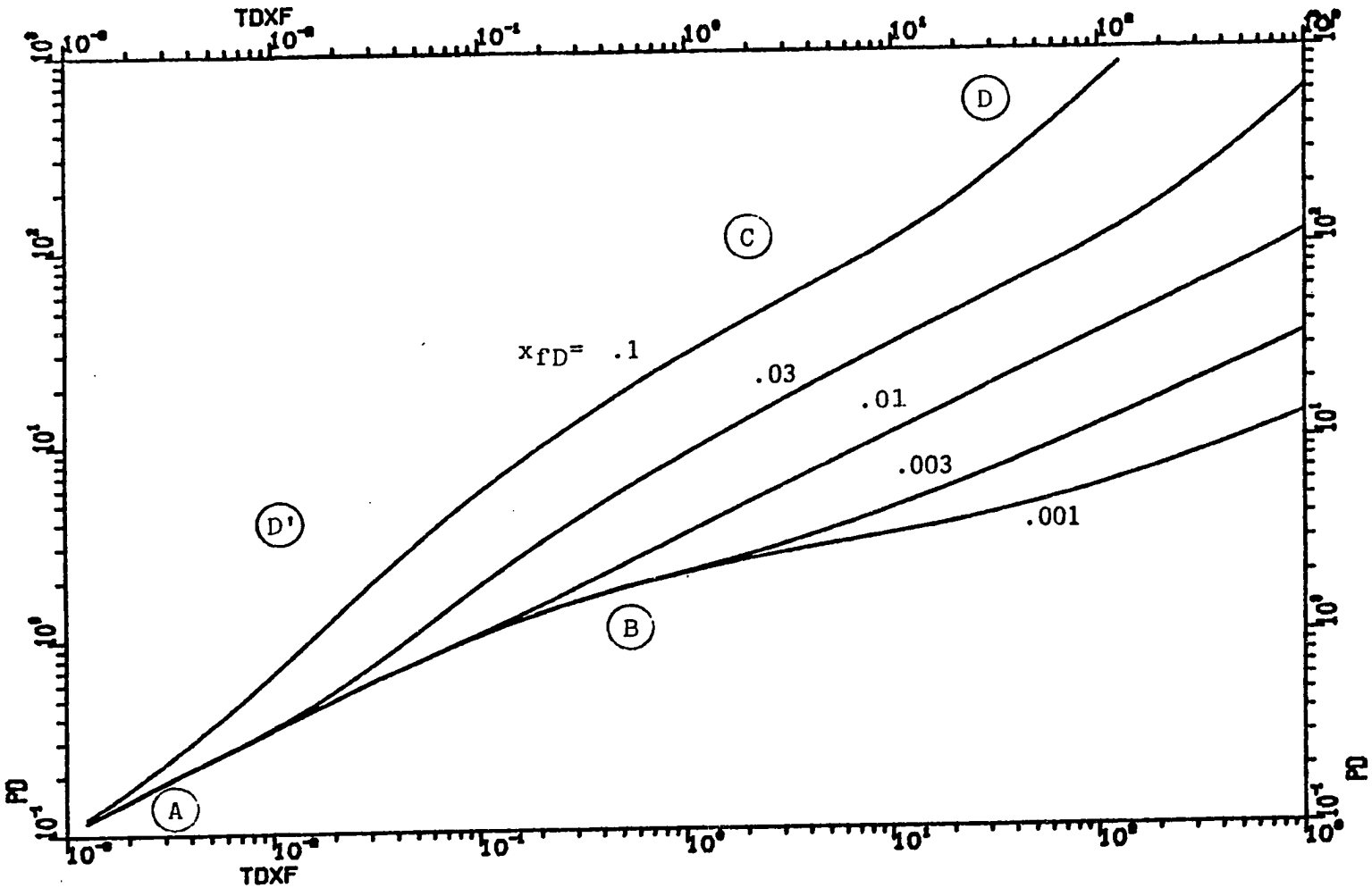


Fig. 4.11. Effect of fracture length.

Gringarten and Ramey (1974) for restricted entry and horizontal fractures. The curvature on the upper pressure curves between the restricted-entry portion and the channel linear portion is a transition effect only. It also appears that a fracture length slightly longer than the width of the channel will show neither pseudo-radial nor restricted-entry effects. A type-curve match for this case would be impossible, since it is the curvature in a plot of the pressure data that demonstrates the flow regimes, and allows a unique match.

Data for a pressure test in a reservoir of the kind described above was simulated by the program. The values of the dimensionless variables used for this case are listed in Table 4.4, and the data are plotted in Fig. 4.12. The program was used in the same fashion as for the geothermal fracture, in order to find a match for the data. The method used is essentially to try and match the early behavior first, then each succeeding flow regime recognizable in the data in later attempts.

In order to match the length of the pseudo-radial portion of the data, the model is used to vary the fracture length, which, as mentioned above, is the controlling factor for this behavior. The results are shown in Fig. 4.13. The value to choose for z_{fD} is .005. By assuming some value of channel length, say 100 units, where the channel width and fracture depth are 1 unit, this value of z_{fD} makes the fracture 1/2 unit long. The next flow regime to match is the transition from channel linear to pseudo-steady-state, and, as

Table 4.3 Values for Dimensionless Variables for Channel Flow Cases.

	channel length	fracture position	fracture length
x_D	.5	.5	.5
y_D	.5	.5,.6,.7,.8,.9	.5
z_D	.5	.5	.5
x_{wD}	.5	.5	.5
y_{wD}	.5	.5,.6,.7,.8,.9	.5
z_{wD}	.5	.5	.5
x_{fD}	1/10,1/20,1/30,1/40	1/100	.1/100,.3/100, 1/100,3/100,10/100
y_{fD}	N/A	N/A	N/A
z_{fD}	1	1	1
R_{yD}	1/10,1/20,1/30,1/40	1/100	1/100
R_{zD}	1/10,1/20,1/30,1/40	1/100	1/100

Table 4.4 Values for Dimensionless Variables for Sample Match Data.

$x_D = .5$	$x_{wD} = .5$	$x_{fD} = .025$	$R_{yD} = .05$
$y_D = .5$	$y_{wD} = .5$	$y_{fD} = N/A$	$R_{zD} = .05$
$z_D = .5$	$z_{wD} = .5$	$z_{fD} = 1$	

mentioned above, the controlling factor for this effect is R_{zD}, the channel length. The channel was assumed to be 80, 40, 20, 10, and 5 units long, and R_{zD} and the dimensionless variables for the x-direction were adjusted accordingly. The results are shown in Fig. 4.14. The match corresponds to an R_{zD} value of 20 units (20 channel widths). Since all of the other variables were well defined for this problem, with the exception of the position of the fracture relative to the sides of the channel, a match was relatively easy to obtain. A much more difficult case to analyze would be the data of Fig. 4.15. The geometry used for this case was a short channel with a long fracture close to one side of the channel. The only clearly defined flow regime is pseudo-steady-state. The small rise in pressure at t_{Dxf} ~.02 might be overlooked in raw data, and thus the channel flow section might be misinterpreted as a continuation of the fracture linear flow. If the pseudo-steady-state section of the curve was missing, a match might not be possible at all. The model can easily establish when such problems can occur by making it convenient to simulate these conditions. A plot such as Fig. 4.15, combined with estimates of the reservoir parameters, would allow one to calculate the approximate start of pseudo-steady-state flow. With this kind of knowledge beforehand, one can make sure that a pressure transient test is performed over a long enough time span to insure the onset of pseudo-steady-state for that well, thus making a later type-curve match more likely, and more unique.

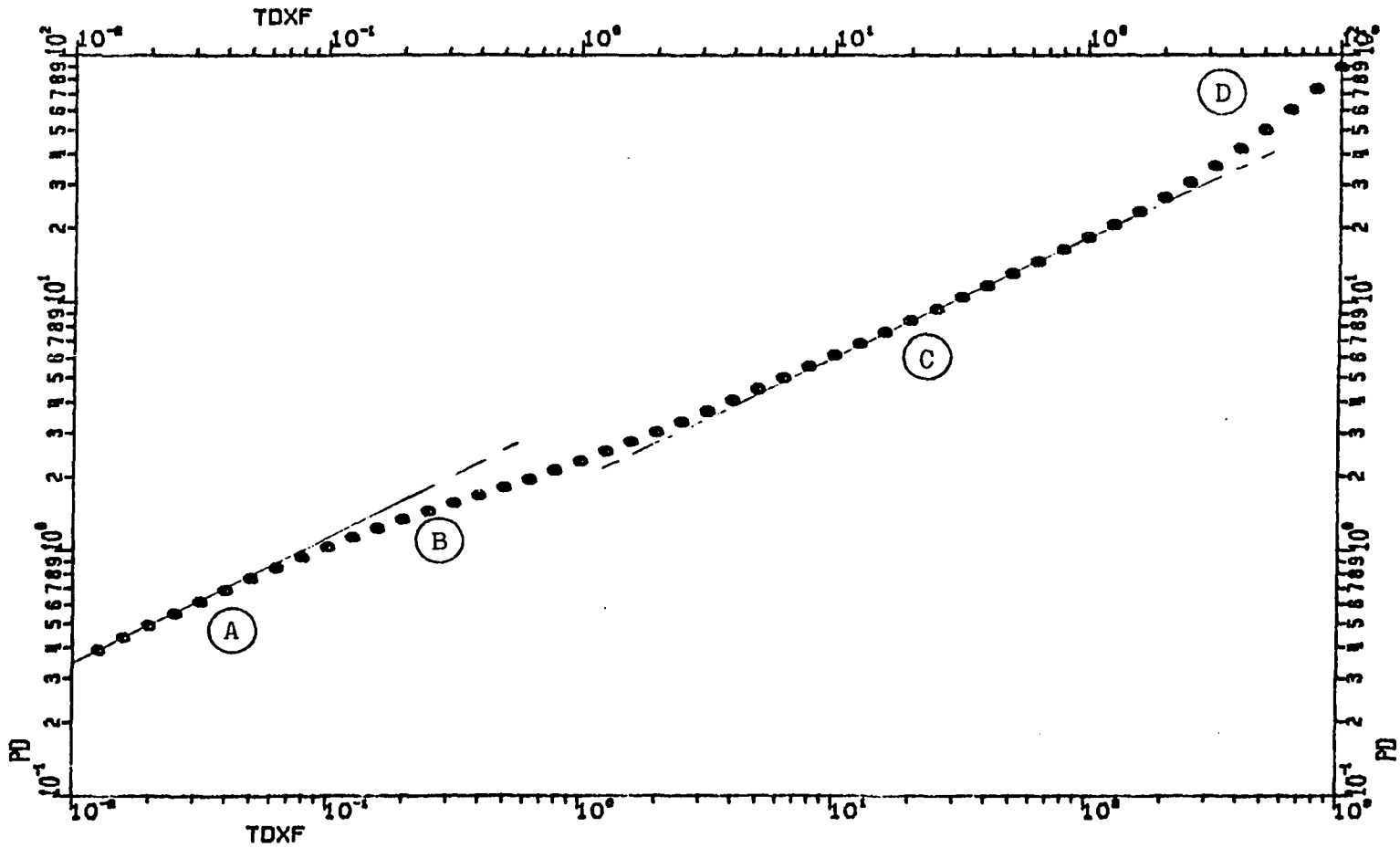


Fig. 4.12. Simulated data for type-curve match demonstration.

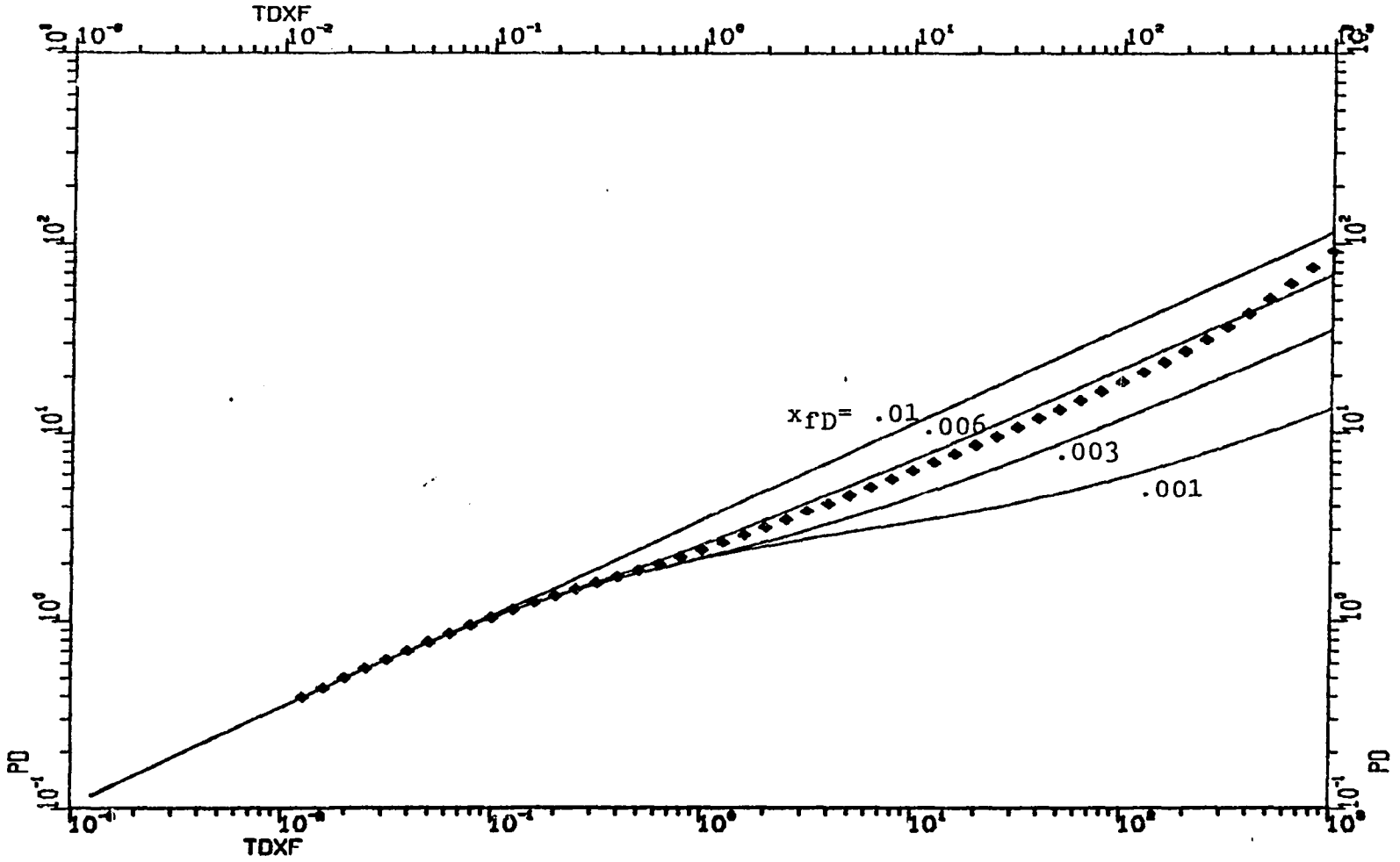


Fig. 4.13. Matching fracture linear and pseudo-radial behaviors.

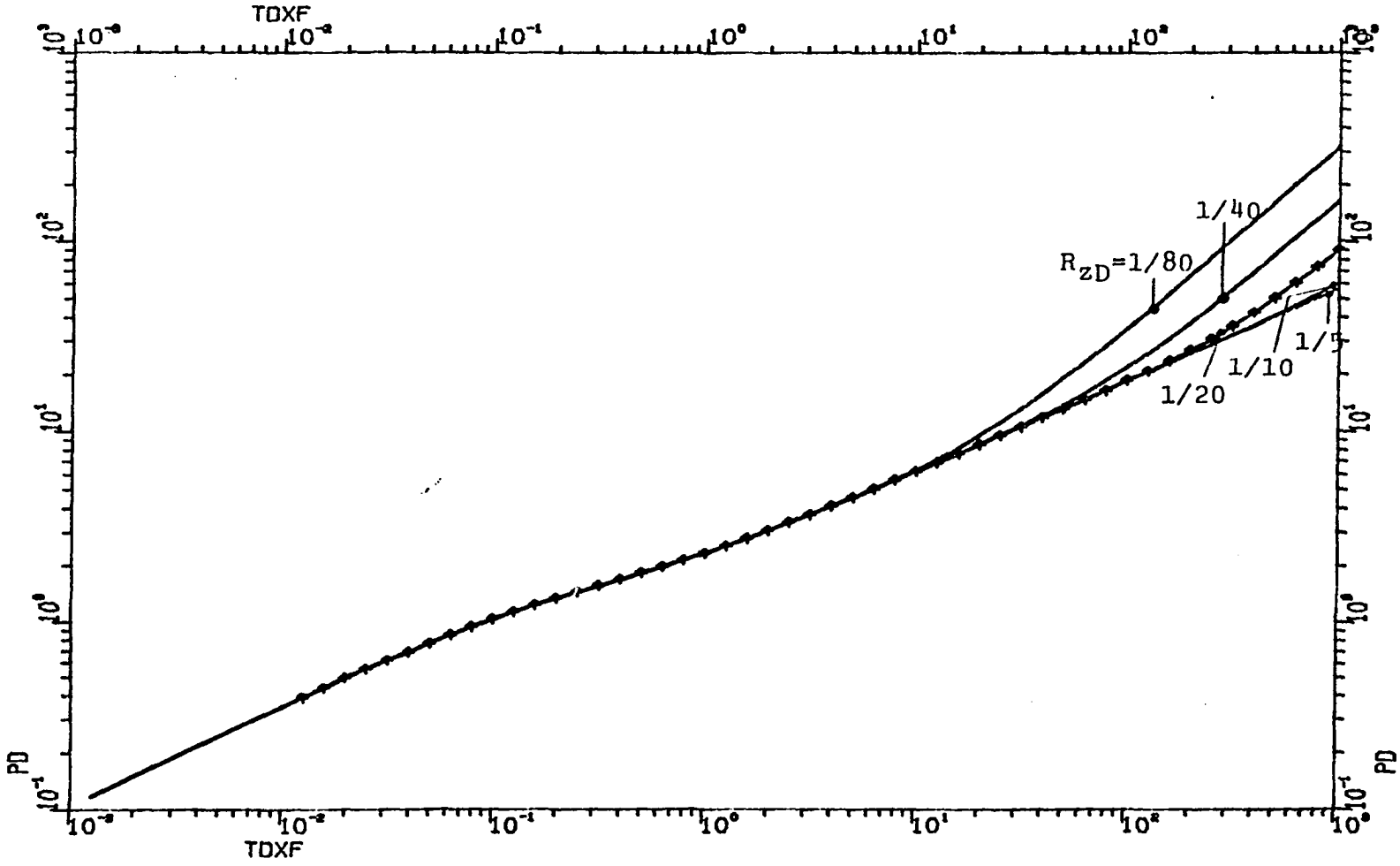


Fig. 4.14. Matching the pseudo-steady-state behavior.

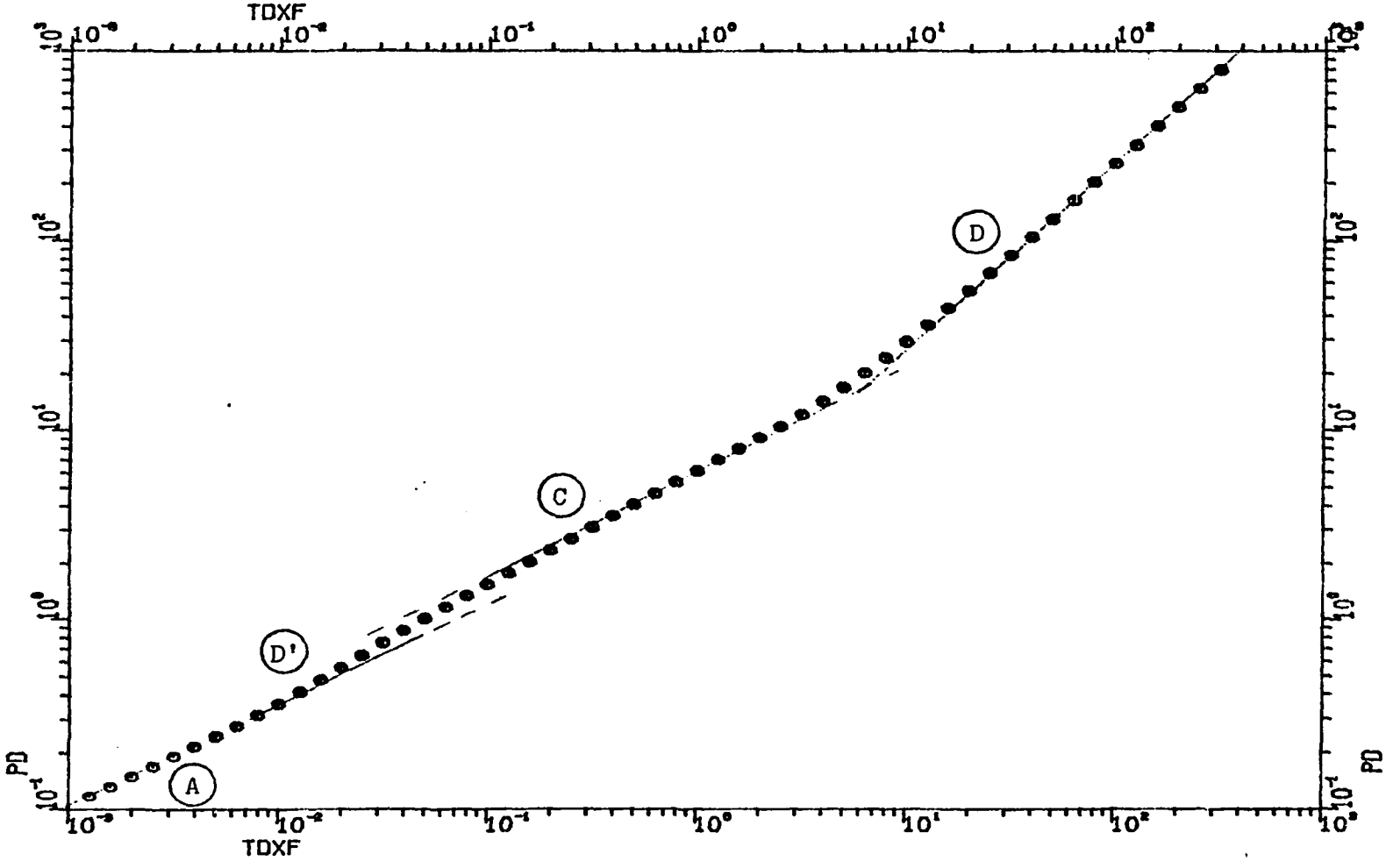


Fig. 4.15. Simulated data for complex channel flow problem.

C. Other Geometries.

The number of other situations that can be modelled is quite large. The computer programs presented in this work allow one the freedom to radically alter a model without resorting to mathematical derivations. It is extremely useful to be able to fix the values of some of the variables to force the model to conform to the known data of a real problem, and to be able to then vary the remaining variables until the pressure response from the model more closely mimics that of the data. Another possible use of the program is in the design of well tests. If reasonable estimates can be made of the reservoir parameters and of the approximate geometry involved for a well, the model can show at what dimensionless times the various flow regimes occur, and via the definition of dimensionless time, at what actual times these regimes will occur. As was mentioned before, it is the appearance of these flow regimes that make unique type-curve matches possible. Therefore, the actual times needed for measurements of pressure for a well test can be estimated, better insuring the possibility of a match, and thus of reliable estimates of reservoir parameters.

CONCLUSIONS

A number of important contributions have been made to the methods used in pressure transient analysis. As evidenced by the large number of papers which provide pressure solutions using the source function approach, the contributions by Gringarten and Ramey (1973) have been very helpful for analyzing a great number of reservoir geometries. By fixing the definitions of the dimensionless variables, two additional benefits are realized. First, the process of combining the source functions together to give an analytic expression has been made very simple, simple enough for a computer to perform this task automatically. Second, the consistency of definitions between solutions for radically different geometries makes qualitative comparisons easier, and aids in the comprehension of the sensitivity of pressure response to geometry changes.

A concise derivation for a new set of source functions has been given. These source functions are most effective when used for early-time calculations. They can be substituted directly for the Laplace forms, which apply to a bounded source. The bounds for the new source functions can be chosen in such a way that unbounded behavior can be simulated, and thus they are capable of being used for a larger class of problems than the Laplace forms by themselves are.

The new source functions are not a replacement for the Laplace versions. Rather, the two forms should be used together, with the new form being used early on, and the Laplace form being used later. A

computational device has been successfully implemented which makes a reasonable compromise between complexity and efficiency.

Practical examples of source functions usage, at both the equation level and at the simplified level dictated by a computer, have been given. The computer program is an especially useful tool, since the equations are manipulated internally, with no additional mathematics or programming necessary. The variables controlling the geometry can be varied at will, to allow one to investigate the many possibilities that the source function approach offers. The program also allows one to approach type-curve matching from a new perspective. The known data can be used immediately to narrow the number of possible geometries down, then the remaining unknowns can be varied one at a time, until a qualitative, then a quantitative, match of the data to a pressure curve can be obtained. More realistic determinations of the sensitivity of reservoir parameter calculations based on type-curve matches can thus be had.

The geothermal fracture example illustrates not only the power of the approach, but also points out the problem of uniqueness of fit and sensitivity. An alternate type-curve match was provided, and significantly different estimates of permeability and depth to boiling interface were calculated. The benefit of the example comes from the knowledge that the method used is an excellent tool for establishing the range of possible interpretations for a given set of data.

For the elongated linear flow system application, the qualitative analysis of a new geometry was presented. The flow regimes possible,

and their effect on pressure response, were illustrated. A sample matching procedure for this case was presented and is shown to be similar to the procedure one would use for any of the geometries which the computer program can simulate.

Another potential use of this method is in the design of pressure transient tests. If the geometry and reservoir parameters can be estimated, a pressure solution not only shows what the response should look like, but also when the flow regimes are established. Thus, it can be concluded that a certain well test requires only several hours, or that another well test requires several days. By better assuring the validity of the well test, the problem of non-uniqueness of fit can be minimized, and the number of well tests necessary might be reduced.

NOMENCLATURE

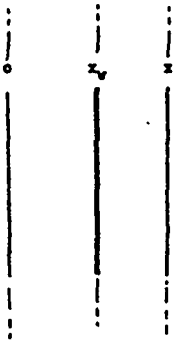
c	total system compressibility, m^2/N
G	Green's function
i	subscript
i_{fD}	x_{fD}, y_{fD}, or z_{fD}; see below
j	subscript
k	peremeability of domain, m^2
n	subscript
$\bar{p}$	product of pressure solutions (see text)
p	pressure at obs. point, N/m^2
p_D	dimensionless pressure
q	volumetric flow rate, m^3/s
r	radial distance, m
r_D	dimensionless radial distance; r/r_w
r_w	wellbore radius, m
R_{yD}	ratio, y_e/x_e
R_{zD}	ratio, z_e/x_e
S	source function
S_D	dimensionless source function
t	time, s
t_D	dimensionless time
t_{Dxf}	dimensionless time; $[\ell_t^2] = x_f^2$ (see text)
$\bar{x}$	position of obs. point; (m,m) radial, (m,m,m) cartesian

$x(,y,z)$	position of obs. point, x-(,y-,z-) direction, m
$x_D(,y_D,z_D)$	dimensionless obs. pos.; $x/x_e(,y/y_e,z/z_e)$
$x_e(,y_e,z_e)$	length of domain, x-(,y-,z-) direction, m
$x_f(,y_f,z_f)$	length of source domain, x-(,y-,z-) direction, m
$x_{fD}(,y_{fD},z_{fD})$	dimensionless source length; $x_f/x_e(,y_f/y_e,z_f/z_e)$
$x_w(,y_w,z_w)$	position of center of source, x-(,y-,z-) direction, m
$x_{wD}(,y_{wD},z_{wD})$	dimensionless source position; $x_w/x_e(,y_w/y_e,z_w/z_e)$
∇	differentiation operator
ε	membership operator
η	diffusivity constant; $\frac{k}{\phi\mu c}$, m^2/s
μ	fluid viscosity, kg/m•s or $N\cdot s/m^2$
ϕ	porosity of domain, fraction
τ	time of start of flow at source, s
τ_D	dimensionless time (dummy variable for integration)

APPENDIX

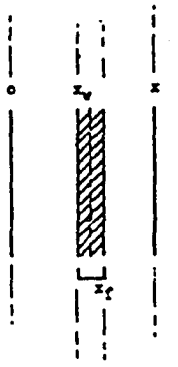
Table A.1 Source Functions.

Plane source



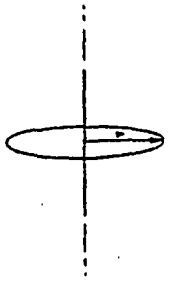
$$I(x) = \exp\left[\frac{-(x-x_w)^2}{4\eta\tau}\right]/2\sqrt{\pi\eta\tau}$$

Slab source



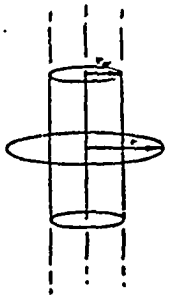
$$II(x) = \frac{1}{2}\left[\operatorname{erf}\left(\frac{x_f/2 + (x-x_w)}{2\sqrt{\eta\tau}}\right) + \operatorname{erf}\left(\frac{x_f/2 - (x_w - x)}{2\sqrt{\eta\tau}}\right)\right]$$

Line source



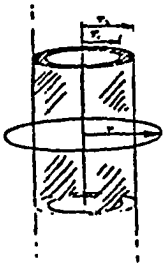
$$III(r) = \exp\left[\frac{-r^2}{4\eta\tau}\right]/4\pi\eta\tau$$

Surface cylinder



$$IV(r) = \frac{r_w}{2\eta\tau} I_o\,(rr_w/2\eta\tau)\,\exp\left[\frac{-(r^2+r_w^2)}{4\eta\tau}\right]$$

Solid cylinder

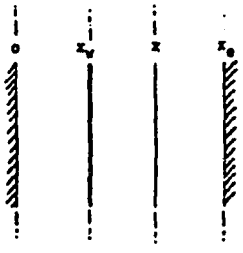


$$V(r) = \exp\left[\frac{-r^2}{4\eta\tau}\right]/2\eta\tau \int_{r_1}^{r_2} I_o(rr'/\eta\tau)\,\exp\left[\frac{-r'^2}{4\eta\tau}\right] r'\,dr'$$

Point source

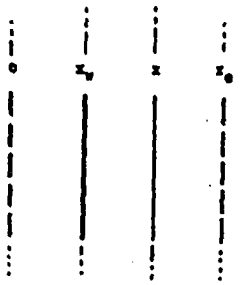
$$VI(d) = \exp\left[\frac{-d^2}{4\eta\tau}\right]/8(\pi\eta\tau)^{3/2},\quad d^2 = (x-x_w)^2 + (y-y_w)^2 + (z-z_w)^2$$

Plane source, no-flow boundaries



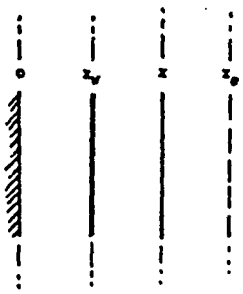
$$VII(x) = \frac{1}{x_e}\left[1+2\sum_{n=1}^{\infty} \exp\left[\frac{-n^2\pi^2\eta\tau}{x_e^2}\right] \cos n\pi\frac{x_w}{x_e}\cos n\pi\frac{x}{x_e}\right]$$

Plane source,
constant-pressure boundaries



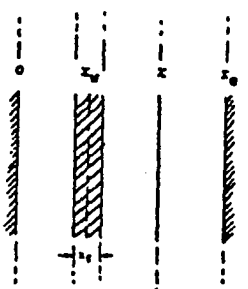
$$VIII(x) = \frac{2}{x_e} \sum_{n=1}^{\infty} \exp\left[\frac{-n^2\pi^2\eta t}{x_e^2}\right] \sin n\pi \frac{x_w}{x_e} \sin n\pi \frac{x}{x_e}$$

Plane source, mixed boundaries



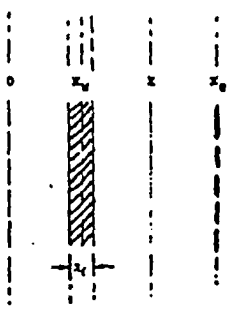
$$IX(x) = \frac{2}{x_e} \sum_{n=1}^{\infty} \exp\left[\frac{-(2n-1)^2\pi^2\eta t}{4x_e^2}\right] \cos (2n-1)\pi \frac{x_w}{x_e} \cos (2n-1)\pi \frac{x}{x_e}$$

Slab source, no-flow boundaries



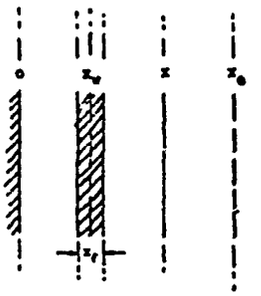
$$X(x) = \left[\frac{x_f}{x_e} + \frac{4}{\pi} \sum_{n=1}^{\infty} \frac{1}{n} \exp\left[\frac{-n^2\pi^2\eta t}{x_e^2}\right] \sin n\frac{\pi}{2}\frac{x_f}{x_e} \cos n\pi \frac{x_w}{x_e} \cos n\pi \frac{x}{x_e}\right]$$

Slab source,
constant-pressure boundaries



$$XI(x) = \frac{4}{\pi} \sum_{n=1}^{\infty} \frac{1}{n} \exp\left[\frac{-n^2\pi^2\eta t}{x_e^2}\right] \sin n\frac{\pi}{2}\frac{x_f}{x_e} \sin n\pi \frac{x_w}{x_e} \sin n\pi \frac{x}{x_e}$$

Slab source, mixed boundaries



$$XII(x) = \frac{8}{\pi} \sum_{n=1}^{\infty} \frac{1}{(2n-1)} \exp\left[\frac{-(2n-1)^2 x^2 \pi^2}{4 x_e^2}\right] \cdot \sin (2n-1) \frac{\pi}{4} \frac{x_f}{x_e} \cos n \frac{\pi}{2} \frac{x_w}{x_e} \cos n \frac{\pi}{2} \frac{x}{x_e}$$

Table A.2 Dimensionless Source Functions.

Radial Forms

$$III_D(r_D) = \frac{1}{r_w^2} \exp(-r_D^2/4\tau_D)/4\pi\tau_D$$

$$IV_D(r_D) = \frac{1}{r_w}\frac{1}{2\tau_D} I_o(r_D/2\tau_D) \exp(-(r_D^2+1)/4\tau_D)$$

$$V_D(r_D) = \exp(-r_D^2/4\tau_D)/2\tau_D \int_{r_{1D}}^{r_{2D}} I_o(r_D r_D'/2\tau_D) \exp(-r_D'^2/4\tau_D) r_D' \, dr_D'$$

$$= \exp(-r_D^2/4\tau_D)/2\tau_D \int_0^1 I_o(r_D r_D'/2\tau_D) \exp(-r_D'^2/4\tau_D) r_D' \, dr_D' \quad *$$

$$r_D = \frac{r}{r_w} \qquad \tau_D = \frac{\eta t}{r_w^2} \qquad r_{1D} = \frac{r_1}{r_w} \qquad r_{2D} = \frac{r_2}{r_w} \qquad r_D' = \frac{r'}{r_w}$$

---

* if $r_1 = 0$

$r_2 = r_w$

Linear Forms

$$VII_D(x_D) = \frac{1}{x_e}\left[1+2\sum_{n=1}^{\infty} \exp(-n^2\pi^2\tau_D)\cos n\pi x_{wD}\cos n\pi x_D\right]$$

$$VIII_D(x_D) = \frac{2}{x_e}\sum_{n=1}^{\infty} \exp(-n^2\pi^2\tau_D)\sin n\pi x_{wD}\sin n\pi x_D$$

$$IX_D(x_D) = \frac{2}{x_e}\sum_{n=1}^{\infty} \exp\left(-\frac{(2n-1)^2}{4}\pi^2\tau_D\right)\cos(2n-1)\pi x_{wD}\cos(2n-1)\pi x_D$$

$$X_D(x_D) = x_{fD} + \frac{4}{\pi}\sum_{n=1}^{\infty} \exp(-n^2\pi^2\tau_D)\sin n\frac{\pi}{2}x_{fD}\cos n\pi x_{wD}\cos n\pi x_D$$

$$XI_D(x_D) = \frac{4}{\pi}\sum_{n=1}^{\infty} \frac{1}{n}\exp(-n^2\pi^2\tau_D)\sin n\frac{\pi}{2}x_{fD}\sin n\pi x_{wD}\sin n\pi x_D$$

$$XII_D(x_D) = \frac{8}{\pi}\sum_{n=1}^{\infty} \frac{1}{(2n-1)}\exp\left(-\frac{(2n-1)^2}{4}\pi^2\tau_D\right)\sin(2n-1)\frac{\pi}{4}x_{fD}$$

$$\cdot\cos(2n-1)\frac{\pi}{2}x_{wD}\cos(2n-1)\frac{\pi}{2}x_D$$

$$VII'_D(x_D) = \frac{1}{x_e}\frac{1}{2\sqrt{\pi\tau_D}}\sum_{j=-\infty}^{+\infty}\left[\exp\left(\frac{-(x_D-(2j+x_{wD}))^2}{4\tau_D}\right)\right.$$

$$\left.+\exp\left(\frac{-(x_D-(2j-x_{wD}))^2}{4\tau_D}\right)\right]$$

$$VIII'_D(x_D) = \frac{1}{x_e}\frac{1}{2\sqrt{\pi\tau_D}}\sum_{j=-\infty}^{\infty}\left[\exp\left(\frac{-(x_D-(2j+x_{wD}))^2}{4\tau_D}\right)\right.$$

$$\left.-\exp\left(\frac{-(x_D-(2j-x_{wD}))^2}{4\tau_D}\right)\right]$$

$$IX'_D(x_D) = \frac{1}{x_e}\frac{1}{2\sqrt{\pi\tau_D}}\sum_{j=-\infty}^{+\infty}(-1)^j\left[\exp\left(\frac{-(x_D-(2j+x_{wD}))^2}{4\tau_D}\right)\right.$$

$$\left.+\exp\left(\frac{-(x_D-(2j-x_{wD}))^2}{4\tau_D}\right)\right]$$

$$X'_D(x_D) = \frac{1}{2}\sum_{j=-\infty}^{+\infty}\left[\operatorname{erf}\left(\frac{x_{fD}/2+(x_D-(2j+x_{wD}))}{2\sqrt{\tau_D}}\right)+\operatorname{erf}\left(\frac{x_{fD}/2-(x_D-(2j+x_{wD}))}{2\sqrt{\tau_D}}\right)\right.$$

$$\left.+\operatorname{erf}\left(\frac{x_{fD}/2+(x_D-(2j-x_{wD}))}{2\sqrt{\tau_D}}\right)+\operatorname{erf}\left(\frac{x_{fD}/2-(x_D-(2j-x_{wD}))}{2\sqrt{\tau_D}}\right)\right]$$

$$XI'_D(x_D) = \frac{1}{2}\sum_{j=-\infty}^{+\infty}\left[\operatorname{erf}\left(\frac{x_{fD}/2+(x_D-(2j+x_{wD}))}{2\sqrt{\tau_D}}\right)+\operatorname{erf}\left(\frac{x_{fD}/2-(x_D-(2j+x_{wD}))}{2\sqrt{\tau_D}}\right)\right.$$

$$\left.-\operatorname{erf}\left(\frac{x_{fD}/2+(x_D-(2j-x_{wD}))}{2\sqrt{\tau_D}}\right)-\operatorname{erf}\left(\frac{x_{fD}/2-(x_D-(2j-x_{wD}))}{2\sqrt{\tau_D}}\right)\right]$$

$$XII'_D(x_D) = \frac{1}{2}\sum_{j=-\infty}^{+\infty}(-1)^j\left[\operatorname{erf}\left(\frac{x_{fD}/2+(x_D-(2j+x_{wD}))}{2\sqrt{\tau_D}}\right)+\operatorname{erf}\left(\frac{x_{fD}/2-(x_D-(2j+x_{wD}))}{2\sqrt{\tau_D}}\right)\right.$$

$$\left.+\operatorname{erf}\left(\frac{x_{fD}/2+(x_D-(2j-x_{wD}))}{2\sqrt{\tau_D}}\right)+\operatorname{erf}\left(\frac{x_{fD}/2-(x_D-(2j-x_{wD}))}{2\sqrt{\tau_D}}\right)\right]$$

$$x_D = \frac{x}{x_e} \qquad x_{wD} = \frac{x_w}{x_e} \qquad x_{fD} = \frac{x_f}{x_e} \qquad \tau_D = \frac{\eta t}{x_e^2}$$

Computer Programs

Each program has been designed to be nested, in the sense that the most important decisions are made first, and the least important ones are made the very last. The most important decision is what the geometry of the desired problem is, so a menu of choices is presented. The radial program prompts for the desired radial geometry first, then for the desired z-axis geometry. If the well is to have any vertical extent at all, one of the slab sources must be chosen for the z-axis, otherwise a point for circle will result.

After a choice is made, the user is prompted for modifications to the dimensionless variables that the program assumes. At this point, One is allowed to effectively convert $[\ell_p]$ and $[\ell_t^2]$ to whatever lengths are desired. For example, the default dimensionless time for the rectangular cases is based on the areal extent of the reservoir (this time is commonly referred to as t_{DA}.) If one is studying a fracture geometry, and the traditional t_{Dxf} dimensionless time is desired, a factor of $x_e y_e / x_f^2$ must be constructed of the available dimensionless length ratios. This would be accomplished by multiplying t_{Da} by the ratio y_e/x_e and then dividing this by x_f/x_e twice. This can be specified easily, as the programs are designed for this type of user interaction.

Next, a decision is made as to whether a drawdown curve or a buildup curve is to be calculated. Since a buildup curve depends on knowing the drawdown pressure response, the drawdown pressures are calculated first, then the superposition identity:

$$p_{Db}[(\Delta t)_D] = p_D[(t_p+\Delta t)_D] - p_D[(\Delta t)_D] \quad (B.1)$$

is used to calculate the buildup pressure.

The values of the dimensionless length ratios are then prompted for. In so doing, the geometry is distorted into some useful configuration, such as a well in a long channel or a fracture near a boundary, and so forth. If a buildup curve is desired, the user is prompted for a time (user's chosen dimensionless units) to use as the flow in time before shut-in. Finally, the range of time over which to calculate the pressure is asked for. A verification of all of the decisions and values that have been established is typed out, and the user is asked whether there are any errors. If there are, a menu of possible actions is provided, and the user chooses whichever one applies.

When a user finds the information given during verification acceptable, the actual calculations take place. The user is occasionally reminded that the program is still at work. By indicating what time period the program is currently working on. Finally, when all calculations are complete, the user is asked whether a tabulation of the results is desired. Meanwhile, the program has already written the results to a file, where they can be manipulated for plotting or other purposes. An option menu is then provided, and the user is allowed to change any or all of the parameters before another set of calculations is performed. Otherwise, the program

terminates. The user is then free to use the result file. A complete example of a program run is given on the following pages.

```
*GO .H
THERE ARE SIX TYPES OF GEOMETRIES AVAILABLE FOR EACH
DIMENSION; PLEASE CHOOSE ANY ONE FOR EACH, SEPARATED BY
COMMAS, LIKE     3,4,3  :
   1. PLANE, NOFLOW, NOFLOW
   2. PLANE, CONSTP, CONSTP
   3. PLANE, NOFLOW, CONSTP
   4.  SLAB, NOFLOW, NOFLOW
   5.  SLAB, CONSTP, CONSTP
   6.  SLAB, NOFLOW, CONSTP
-----------------WHICH TYPES FOR X,Y,Z?
=4,1,4
IS THIS DRAWDOWN (0) OR BUILDUP (1) ?
=0
THE DIMENSIONLESS VARIABLES ARE:
   1. XD   2. YD   3. ZD
   4. XWD  5. YWD  6. ZWD
   7. XFD  8. YFD  9. ZFD
    10. YE/XE 11. ZE/XE
YOU MAY MULTIPLY OR DIVIDE THE DEFAULT VALUES FOR PD
AND TD BY ANY OF THESE TO CREATE THE TD AND PD OF YOUR CHOICE.
TD DEFAULTS TO:
                      K * T
TD = TDA = ----------------------
           PHI * MU * CT * XE * YE
AND PD DEFAULTS TO:
     2 * PI * K * ZE * DELTA-P
PD = ------------------------
            Q * MU
WHAT TO MULTIPLY TD BY (0 TO QUIT)?
=10
WHAT TO MULTIPLY TD BY (0 TO QUIT)?
=0
WHAT TO DIVIDE TD BY (0 TO QUIT)?
=7
WHAT TO DIVIDE TD BY (0 TO QUIT)?
=7
WHAT TO DIVIDE TD BY (0 TO QUIT)?
=0
WHAT TO MULTIPLY PD BY (0 TO QUIT)?
=0
WHAT TO DIVIDE PD BY (0 TO QUIT)?
=0
WHEN ENTERING VALUES FOR THE DIMENSIONLESS VARIABLES TO
FOLLOW, ENTER 0 FOR THOSE THAT DO NOT APPLY TO THE PROBLEM:
WHAT ARE XD,YD,ZD?
=.5,.5,.5
WHAT ARE XWD,YWD,ZWD?
=.5,.5,.5
WHAT ARE XFD,YFD,ZFD?
```

```
=.025,1,1
WHAT ARE YE/XE, ZE/XE?
=.05,.05
FOR THE TIME RANGE, PLEASE USE LOG-CYCLE NOTATION:
FOR A RANGE OF TD=10**-3 TO 10**2, ENTER -3,2.
WHAT (BEGIN,END) CYCLES?
=-2,3
LOWER AND UPPER LIMITS ON PD (YOUR UNITS)?
=0,1000
**** MODEL VERIFICATION ****
X-AXIS GEOMETRY IS TYPE                4
Y-AXIS GEOMETRY IS TYPE                1
Z-AXIS GEOMETRY IS TYPE                4
---TD HAS BEEN MODIFIED.
XD,YD,ZD     =
 0.50000D 00 0.50000D 00 0.50000D 00
XWD,YWD,ZWD  =
 0.50000D 00 0.50000D 00 0.50000D 00
XFD,YFD,ZFD  =
 0.25000D-01 0.10000D 01 0.10000D 01
YE/XE,ZE/XE  =
 0.50000D-01 0.50000D-01
START,END CYCLES ARE SET TO :              -2              3
LOWER LIMIT OF PD =  0.
UPPER LIMIT OF PD =  0.1000000000000000000D 04
========== IS EVERYTHING OKAY (1=YES)?
=1
NOW CALCULATING FOR LOG CYCLE              -4
NOW CALCULATING FOR LOG CYCLE              -3
NOW CALCULATING FOR LOG CYCLE              -2
NOW CALCULATING FOR LOG CYCLE              -1
NOW CALCULATING FOR LOG CYCLE               0
   EXP UFL AT LOCATION 007674

   EXP UFL AT LOCATION 007674

   EXP UFL AT LOCATION 007674

   EXP UFL AT LOCATION 007674

   EXP UFL AT LOCATION 007674

   EXP UFL AT LOCATION 007674

**THIS IS THE LAST TIME THE ABOVE MESSAGE WILL APPEAR**
NOW CALCULATING FOR LOG CYCLE               1
NOW CALCULATING FOR LOG CYCLE               2
NOW WRITING TD AND PD TO TEMP FILE 13.
              5              5
             10              5
```

```
                5                5
DO YOU WISH TO SEE A TABLE (1=YES)?
=1
    TD         PD         TD         PD         TD         PD
==========================================================================
0.100E-01 0.328E 00 0.501E 00 0.179E 01 0.251E 02 0.938E 01
0.126E-01 0.371E 00 0.631E 00 0.194E 01 0.316E 02 0.105E 02
0.158E-01 0.419E 00 0.794E 00 0.211E 01 0.398E 02 0.117E 02
0.200E-01 0.473E 00 0.100E 01 0.230E 01 0.501E 02 0.130E 02
0.251E-01 0.533E 00 0.126E 01 0.251E 01 0.631E 02 0.146E 02
0.316E-01 0.599E 00 0.158E 01 0.275E 01 0.794E 02 0.163E 02
0.398E-01 0.672E 00 0.200E 01 0.302E 01 0.100E 03 0.183E 02
0.501E-01 0.750E 00 0.251E 01 0.332E 01 0.126E 03 0.206E 02
0.631E-01 0.833E 00 0.316E 01 0.366E 01 0.158E 03 0.233E 02
0.794E-01 0.922E 00 0.398E 01 0.404E 01 0.200E 03 0.266E 02
0.100E 00 0.101E 01 0.501E 01 0.447E 01 0.251E 03 0.307E 02
0.126E 00 0.111E 01 0.631E 01 0.495E 01 0.316E 03 0.358E 02
0.158E 00 0.121E 01 0.794E 01 0.550E 01 0.398E 03 0.422E 02
0.200E 00 0.132E 01 0.100E 02 0.610E 01 0.501E 03 0.503E 02
0.251E 00 0.142E 01 0.126E 02 0.679E 01 0.631E 03 0.605E 02
0.316E 00 0.154E 01 0.158E 02 0.755E 01 0.794E 03 0.734E 02
0.398E 00 0.166E 01 0.200E 02 0.841E 01 0.100E 04 0.895E 02
OPTIONS:
  1. NEW CYCLE RANGE FOR TIME.
  2. NEW VALUES FOR DIMENSIONLESS VARIABLES.
  3. NEW DEFINITIONS FOR TD AND/OR PD.
  4. NEW MODEL GEOMETRY.
  5. QUIT.
======== WHICH ONE?
=5
```

```
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION TD(200),PD(200),DV(5),ITP(5),ITQ(5),IPP(5),IPQ(5),F(11)
      DIMENSION DT(11),PDB(200),ICNT(2)
      LOGICAL ITGL
      COMMON/P1VARS/DV,SR,SZ,TR,TZ,PI
      NTP=0
      NTQ=0
      NPP=0
      NPQ=0
      PI=4D0*DATAN(1D0)
      DT(1)=1D0
      DDT=10D0**1D-2
      DO 5 I=2,11
    5 DT(I)=DT(I-1)*DDT
   10 PRINT,'WHICH TYPE OF RADIAL SYSTEM:'
      PRINT,'  1. LINE SOURCE'
      PRINT,'  2. SURFACE CYLINDER SOURCE'
      PRINT,'  3. SOLID CYLINDER SOURCE'
      READ,IRT
      IF((IRT.GT.3).OR.(IRT.LT.1))GO TO 10
   15 PRINT,'WHICH TYPE OF VERTICAL SYSTEM:'
      PRINT,'  1. PLANE SOURCE, NOFLOW, NOFLOW'
      PRINT,'  2. PLANE SOURCE, CONST.P. , CONST.P.'
      PRINT,'  3. PLANE SOURCE, NOFLOW, CONST.P.'
      PRINT,'  4. SLAB SOURCE, NOFLOW, NOFLOW'
      PRINT,'  5. SLAB SOURCE, CONST.P. , CONST.P.'
      PRINT,'  6. SLAB SOURCE, NOFLOW, CONST.P.'
      READ,IZT
      IF((IZT.LT.1).OR.(IZT.GT.6))GO TO 15
      PRINT,'IS THIS DRAWDOWN (0) OR BUILDUP (1)?'
      READ,IBLDP
   20 PRINT,'DUE TO GEOMETRY, TD ALWAYS EQUALS:'
      PRINT,'        K * T'
      PRINT,'----------------------'
      PRINT,'PHI * MU * CT * RW**2'
      IF(NTP+NTQ.GT.0)PRINT,'......YOU HAVE ALREADY MODIFIED TD'
      PRINT,'DIMENSIONLESS VARIABLES:'
      PRINT,'  1. RD   = R/RW'
      PRINT,'  2. ZWD  = ZW/ZE'
      PRINT,'  3. ZFD  = ZF/ZE'
      PRINT,'  4. ZD   = Z/ZE'
      PRINT,'  5. ZERW = ZE/RW'
      PRINT,'YOU MAY MULTIPLY OR DIVIDE BY ONE OR MORE OF THESE.'
      PRINT,'IS TD OKAY (1=YES)?'
      READ,I
      IF(I.EQ.1)GO TO 30
      NTP=0
   21 PRINT,'WHICH WOULD YOU MULTIPLY BY? (ENTER 0 IF DONE)'
      READ,I
      IF((I.LT.1).OR.(I.GT.5))GO TO 22
```

```
      NTP=NTP+1
      ITP(NTP)=I
      GO TO 21
   22 NTQ=0
   23 PRINT,'WHICH WOULD YOU DIVIDE BY? (0 IF DONE)'
      READ,I
      IF((I.LT.1).OR.(I.GT.5))GO TO 30
      NTQ=NTQ+1
      ITQ(NTQ)=I
      GO TO 23
   30 PRINT,'ALSO BECAUSE OF GEOMETRY, PD ALWAYS EQUALS:'
      PRINT,'2 * PI * K * ZE * DELTA-P'
      PRINT,'------------------------'
      PRINT,'         Q * MU'
      IF(NPP+NPQ.GT.0)PRINT,'.....YOU HAVE ALREADY MODIFIED PD.'
      PRINT,'IS PD OKAY? (1=YES)'
      READ,I
      IF(I.EQ.1)GO TO 40
      PRINT,'USING THE SAME NUMBERING FOR THE DIMENSIONLESS VARIABLES,'
      NPP=0
   31 PRINT,'WHICH WOULD YOU MULTIPLY BY? (0 TO QUIT)'
      READ,I
      IF((I.LT.1).OR.(I.GT.5))GO TO 32
      NPP=NPP+1
      IPP(NPP)=I
      GO TO 31
   32 NPQ=0
   33 PRINT,'WHICH WOULD YOU DIVIDE BY? (0 TO QUIT)'
      READ,I
      IF((I.LT.1).OR.(I.GT.5))GO TO 40
      NPQ=NPQ+1
      IPQ(NPQ)=I
      GO TO 33
   40 PRINT,'WHAT ARE THE VALUES OF RD,ZWD,ZFD,ZD,ZERW ?'
      PRINT,'(ZFD CANNOT BE ZERO IF YOU USE A SLAB SOURCE VERTICALLY)'
      READ,DV(1),DV(2),DV(3),DV(4),DV(5)
      IF(IBLDP.EQ.0)GO TO 49
      PRINT,'WHAT IS THE SHUT-IN TIME (YOUR UNITS)?'
      READ,TDP
   49 FT=1D0
      IF(NTP.EQ.0)GO TO 42
      DO 41 I=1,NTP
   41 FT=FT/DV(ITP(I))
   42 IF(NTQ.EQ.0)GO TO 44
      DO 43 I=1,NTQ
   43 FT=FT*DV(ITQ(I))
   44 FP=1D0
      IF(NPP.EQ.0)GO TO 46
      DO 45 I=1,NPP
   45 FP=FP*DV(IPP(I))
```

```
46 IF(NPQ.EQ.0)GO TO 50
   DO 47 I=1,NPQ
47 FP=FP/DV(IPQ(I))
50 PRINT,'WHAT TIME RANGE?'
   PRINT,'(PLEASE USE POWERS-OF-TEN NOTATION:'
   PRINT,' IF YOU WANT TIME TO START AT 10**(-4), ENTER -4 FOR START.'
   PRINT,' IF YOU WANT TIME TO END AT 10**(3), ENTER 3 FOR END.)'
   PRINT,'WHEN TO START,END?'
   READ,NCI,NCF
   PRINT,'WHAT IS THE LOWER LIMIT ON PD (YOUR UNITS)?'
   READ,PDLOW
   PRINT,'WHAT IS THE UPPER LIMIT ON PD (YOUR UNITS)?'
   READ,PDHI
   PRINT,'******* VERIFICATION *******'
   PRINT,'RADIAL SYSTEM =',IRT
   PRINT,'Z-AXIS SYSTEM =',IZT
   IF(NTP.GT.0)PRINT,'TD MODIFIED BY',NTP,' PRODUCT TERMS'
   IF(NTQ.GT.0)PRINT,'TD MODIFIED BY',NTQ,' QUOTIENT TERMS'
   IF(NPP.GT.0)PRINT,'PD MODIFIED BY',NPP,' PRODUCT TERMS'
   IF(NPQ.GT.0)PRINT,'PD MODIFIED BY',NPQ,' QUOTIENT TERMS'
   PRINT,'RD   =',DV(1)
   PRINT,'ZWD  =',DV(2)
   PRINT,'ZFD  =',DV(3)
   PRINT,'ZD   =',DV(4)
   PRINT,'ZERW =',DV(5)
   PRINT,'START CYCLE =',NCI
   PRINT,'  END CYCLE =',NCF
   PRINT,'LOWER PD LIMIT =',PDLOW
   PRINT,'UPPER PD LIMIT =',PDHI
   PRINT,'.......IS EVERYTHING OKAY? (1=YES)'
   READ,I
   IF(I.NE.1)GO TO 60
   IF(IRT.EQ.1)FA=PI*FP*FT
   IF(IRT.EQ.2)FA=FP*FT/2D0
   IF(IRT.EQ.3)FA=FP*FT
   IF(IZT.GT.3)FA=FA/DV(3)
   NCI=NCI-1
   NC=NCF-NCI
   TD(1)=0D0
   PD(1)=0D0
   I=1
   TDU=10D0**NCI
   F(1)=0D0
   ICNT(1)=5
   ICNT(2)=5
   ITGL=.FALSE.
   DO 53 NCC=1,NC
   PRINT,'NOW CALCULATING PD FOR LOG CYCLE',NCI+NCC-1
   TDL=TDU
   DO 52 NL=1,10
```

```
      DO 51 J=2,11
      T=FT*TDL*DT(J)
      TR=T
      TZ=TR/DV(5)/DV(5)
      IF(IRT.EQ.1)CALL RLINE
      IF(IRT.EQ.2)CALL RCYL
      IF(IRT.EQ.3)CALL RSOLID
      IF(IZT.EQ.1)CALL ZPNN(ITGL,ICNT)
      IF(IZT.EQ.2)CALL ZPCC(ITGL,ICNT)
      IF(IZT.EQ.3)CALL ZPNC(ITGL,ICNT)
      IF(IZT.EQ.4)CALL ZSNN(ITGL,ICNT)
      IF(IZT.EQ.5)CALL ZSCC(ITGL,ICNT)
      IF(IZT.EQ.6)CALL ZSNC(ITGL,ICNT)
   51 F(J)=SR*SZ
      I=I+1
      TD(I)=TDL*DT(11)
      PD(I)=PD(I-1)
      DO 55 J=2,11
   55 PD(I)=PD(I)+(F(J)+F(J-1))*FA*TDL*(DT(J)-DT(J-1))
      IF(PD(I).GE.PDLOW)GO TO 400
      I=I-1
      PD(I)=PD(I+1)
      TD(I)=TD(I+1)
      GO TO 401
  400 IF(PD(I).LE.PDHI)GO TO 401
      I=I-1
  401 F(1)=F(11)
   52 TDL=TDL*DT(11)
   53 TDU=TDU*10D0
      IF(IBLDP.EQ.0)GO TO 56
      PDB(1)=0D0
      J=1
  201 J=J+1
      T=TDP+TD(J)
      IL=1
      IH=I
  202 K=(IL+IH)/2
      IF(TD(K).GT.T)GO TO 203
      IL=K
      IF((IL+1).EQ.IH)GO TO 204
      GO TO 202
  203 IH=K
      IF((IH-1).EQ.IL)GO TO 204
      GO TO 202
  204 PDI=PD(IL)+(T-TD(IL))*(PD(IH)-PD(IL))/(TD(IH)-TD(IL))
      PDB(J)=PDI-PD(J)
      IF(T.LT.TD(I))GO TO 201
   56 PRINT,'NOW WRITING TD AND PD TO TEMP FILE 13.'
      NP=I-1
      IF(IBLDP.EQ.1)NP=J-1
```

```
      WRITE(13,1000)NP
      IF(IBLDP.EQ.1)GO TO 58
      WRITE(13,1100)(TD(J),PD(J),J=2,I)
      GO TO 59
   58 WRITE(13,1100)(TD(KK),PDB(KK),KK=2,J)
 1000 FORMAT(I3)
 1100 FORMAT(2E12.4)
   59 PRINT,'DO YOU WISH TO SEE A TABLE? (1=YES)'
      READ,K
      IF(K.NE.1)GO TO 60
      IF(IBLDP.EQ.1)I=J
      NR=(I+2)/3
      I1=NR
      I2=2*NR
      PRINT,'    TD         PD         TD         PD         TD         PD'
      PRINT,'=============================================================='
      IF(IBLDP.EQ.1)GO TO 300
      DO 54 J=1,NR
   54 WRITE(6,1200)TD(J),PD(J),TD(J+I1),PD(J+I1),TD(J+I2),PD(J+I2)
      GO TO 60
 1200 FORMAT(6E10.3)
  300 DO 57 J=1,NR
   57 WRITE(6,1200)TD(J),PDB(J),TD(J+I1),PDB(J+I1),TD(J+I2),PDB(J+I2)
   60 PRINT,'OPTIONS:'
      PRINT,'  1. NEW TIME RANGE'
      PRINT,'  2. NEW VALUES FOR DIMENSIONLESS VARIABLES'
      PRINT,'  3. NEW DEFINITIONS FOR TD AND/OR PD'
      PRINT,'  4. NEW MODEL GEOMETRY'
      PRINT,'  5. QUIT'
      PRINT,'WHICH ONE?'
      READ,J
      IF(J.LT.1)GO TO 60
      IF(J.GT.5)GO TO 60
      GO TO (50,40,20,10,70),J
   70 STOP
      END
      SUBROUTINE RLINE
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/P1VARS/DV(5),SR,SZ,TR,TZ,PI
      SR=DDEXP(-DV(1)*DV(1)/(4D0*TR))/(4D0*PI*TR)
      RETURN
      END
      SUBROUTINE RCYL
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/P1VARS/DV(5),SR,SZ,TR,TZ,PI
      ARG1=DV(1)/2D0/TR
      ARG2=-.25*(DV(1)*DV(1)+1)/TR
      SR=.5D0*BI0EXP(ARG1,ARG2)/TR
      RETURN
      END
```

```
      SUBROUTINE RSOLID
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/P1VARS/DV(5),SR,SZ,TR,TZ,PI
      DIMENSION F(21)
      IF(DABS(DV(1)-1D0).LT.1D-4)GO TO 5
      IF(DV(1).GT.1D0)GO TO 7
      IF(DV(1).LT.1D0)GO TO 6
    5 IF(TR.GT.(3.1416D-4))GO TO 9
      SR=.5D0
      RETURN
    6 DEL=1D0-DV(1)
      IF(TR.GT.(5D-2*DEL*DEL))GO TO 9
      SR=1D0
      RETURN
    7 DEL=DV(1)-1D0
      IF(TR.GT.(5D-2*DEL*DEL))GO TO 9
      SR=0D0
      RETURN
    9 C1=.5D0*DV(1)/TR
      C2=.25D0/TR
      C3=-DV(1)*DV(1)*C2
      F(1)=0D0
      R=0D0
      DO 10 J=2,21
      R=R+.5D-1
   10 F(J)=R*BI0EXP(C1*R,C3-C2*R*R)
      SR=F(1)+F(21)
      DO 20 J=2,20,2
   20 SR=SR+4D0*F(J)
      DO 30 J=3,19,2
   30 SR=SR+2D0*F(J)
      SR=SR/TR/120D0
      RETURN
      END
      DOUBLE PRECISION FUNCTION BI0EXP(ARG1,ARG2)
      IMPLICIT REAL*8 (A-Z)
      IF(ARG1.GT.3.75D0)GO TO 10
      BI0EXP=BESI01(ARG1)*DDEXP(ARG2)
      RETURN
   10 ARG3=ARG1+ARG2
      BI0EXP=BESI02(ARG1,ARG3)
      RETURN
      END
      DOUBLE PRECISION FUNCTION DDEXP(ARG)
      IMPLICIT REAL*8 (A-Z)
      IF(DABS(ARG).LE.87D0)GO TO 20
      IF(ARG.GT.87D0)GO TO 10
      DDEXP=0D0
      RETURN
   10 DDEXP=1D38
```

```
      RETURN
   20 DDEXP=DEXP(ARG)
      RETURN
      END
      DOUBLE PRECISION FUNCTION BESI01(ARG)
      IMPLICIT REAL*8 (A-Z)
      DIMENSION C(6)
      DATA C/3.5156229D0,3.0899424D0,1.2067492D0,.2659732D0,
     &.0360768D0,.0045813D0/
      T=ARG/3.75D0
      BESI01=((C(6)*T*T+C(5))*T*T+C(4))*T*T+C(3)
      BESI01=((BESI01*T*T+C(2))*T*T+C(1))*T*T+1D0
      RETURN
      END
      DOUBLE PRECISION FUNCTION BESI02(ARG1,ARG2)
      IMPLICIT REAL*8 (A-Z)
      DIMENSION C(9)
      DATA C/.39894228D0,.01328592D0,.000225319D0,-.00157565D0,
     &.00916281D0,-.02057706D0,.026355337D0,-.01647633D0,.000392377D0/
      T=ARG1/3.75D0
      BESI02=(((C(9)/T+C(8))/T+C(7))/T+C(6))/T+C(5)
      BESI02=(((BESI02/T+C(4))/T+C(3))/T+C(2))/T+C(1)
      BESI02=BESI02*DDEXP(ARG2)/DSQRT(ARG1)
      RETURN
      END
      SUBROUTINE ZPNN(ITGL,ICNT)
      INTEGER ICNT(2)
      LOGICAL ITGL
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/P1VARS/DV(5),SR,SZ,TR,TZ,PI
      IF(ITGL)GO TO 100
      C1=DV(4)-DV(2)
      C2=DV(4)+DV(2)
      C3=.25D0/(TZ)
      SU=DDEXP(-C1*C1*C3)+DDEXP(-C2*C2*C3)
      DO 10 J=1,100
      F=DDEXP(-(C1-2*J)**2*C3)+DDEXP(-(C2-2*J)**2*C3)
      F=F+DDEXP(-(C1+2*J)**2*C3)+DDEXP(-(C2+2*J)**2*C3)
      IF(F.LT.1D-6*SU)GO TO 40
   10 SU=SU+F
   40 SZ=SU/2D0/DSQRT(PI*TZ)
      IF(J.LT.ICNT(1))RETURN
      ITGL=.TRUE.
      ICNT(1)=ICNT(1)+5
      RETURN
  100 SU=1D0
      C1=PI*PI*TZ
      C2=PI*DV(2)
      C3=PI*DV(4)
      DO 110 J=1,200
```

```
      F=2D0*DDEXP(-J*J*C1)
      IF(F.LT.1D-6*DABS(SU))GO TO 120
  110 SU=SU+F*DCOS(J*C2)*DCOS(J*C3)
  120 SZ=SU
      IF(J.LT.ICNT(2))RETURN
      ITGL=.FALSE.
      ICNT(2)=ICNT(2)+5
      RETURN
      END
      SUBROUTINE ZPCC(ITGL,ICNT)
      INTEGER ICNT(2)
      LOGICAL ITGL
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/P1VARS/DV(5),SR,SZ,TR,TZ,PI
      IF(ITGL)GO TO 100
      C1=DV(4)-DV(2)
      C2=DV(4)+DV(2)
      C3=.25D0/(TZ)
      SU=DDEXP(-C1*C1*C3)-DDEXP(-C2*C2*C3)
      DO 10 J=1,100
      F=DDEXP(-(C1-2*J)**2*C3)-DDEXP(-(C2-2*J)**2*C3)
      F=F+DDEXP(-(C1+2*J)**2*C3)-DDEXP(-(C2+2*J)**2*C3)
      IF(F.LT.1D-6*SU)GO TO 40
   10 SU=SU+F
   40 SZ=SU/2D0/DSQRT(PI*TZ)
      IF(J.LT.ICNT(1))RETURN
      ICNT(1)=ICNT(1)+5
      ITGL=.TRUE.
      RETURN
  100 SU=0D0
      C1=PI*PI*TZ
      C2=PI*DV(2)
      C3=PI*DV(4)
      DO 110 J=1,200
      F=2D0*DDEXP(-J*J*C1)
      IF(F.LT.1D-6*DABS(SU))GO TO 120
  110 SU=SU+F*DSIN(J*C2)*DSIN(J*C3)
  120 SZ=SU
      IF(J.LT.ICNT(2))RETURN
      ICNT(2)=ICNT(2)+5
      ITGL=.FALSE.
      RETURN
      END
      SUBROUTINE ZPNC(ITGL,ICNT)
      INTEGER ICNT(2)
      LOGICAL ITGL
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/P1VARS/DV(5),SR,SZ,TR,TZ,PI
      IF(ITGL)GO TO 100
      C1=DV(4)-DV(2)
```

```
      C2=DV(4)+DV(2)
      C3=.25D0/(TZ)
      SIGN=-1D0
      SU=DDEXP(-C1*C1*C3)+DDEXP(-C2*C2*C3)
      DO 10 J=1,100
      F=DDEXP(-(C1-2*J)**2*C3)+DDEXP(-(C2-2*J)**2*C3)
      F=F+DDEXP(-(C1+2*J)**2*C3)+DDEXP(-(C2+2*J)**2*C3)
      IF(F.LT.1D-6*SU)GO TO 40
      SU=SU+SIGN*F
      IF(DABS(SU).LT.1D-6)GO TO 40
   10 SIGN=-SIGN
   40 SZ=SU/2D0/DSQRT(PI*TZ)
      IF(J.LT.ICNT(1))RETURN
      ICNT(1)=ICNT(1)+5
      ITGL=.TRUE.
      RETURN
  100 SU=0D0
      C1=PI*PI*TZ*.25D0
      C2=PI*DV(2)
      C3=PI*DV(4)
      DO 110 J=1,200
      J1=2*J+1
      F=2D0*DDEXP(-J1*J1*C1)
      IF(F.LT.1D-6*DABS(SU))GO TO 120
      SU=SU+F*DCOS(J1*C2)*DCOS(J1*C3)
      IF(DABS(SU).LT.1D-6)GO TO 120
  110 CONTINUE
  120 SZ=SU
      IF(J.LT.ICNT(2))RETURN
      ICNT(2)=ICNT(2)+5
      ITGL=.FALSE.
      RETURN
      END
      SUBROUTINE ZSNN(ITGL,ICNT)
      INTEGER ICNT(2)
      LOGICAL ITGL
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/P1VARS/DV(5),SR,SZ,TR,TZ,PI
      IF(ITGL)GO TO 100
      IF(DV(3).LT.0.99D0)GO TO 5
      SU=2D0
      GO TO 40
    5 C=.5D0*DV(3)
      CB=.5D0/DSQRT(TZ)
      CC=2D0*CB
      C1=CB*(C+DV(4)-DV(2))
      C2=CB*(C-DV(4)+DV(2))
      C3=CB*(C+DV(4)+DV(2))
      C4=CB*(C-DV(4)-DV(2))
      SU=DERF(C1)+DERF(C2)+DERF(C3)+DERF(C4)
```

```
      N=(C3+10D0)/CC
      IF(N.LE.1)GO TO 40
      DO 10 J=1,100
      F=DERF(C1-J*CC)+DERF(C2-J*CC)+DERF(C3-J*CC)+DERF(C4-J*CC)
      F=F+DERF(C1+J*CC)+DERF(C2+J*CC)+DERF(C3+J*CC)+DERF(C4+J*CC)
      IF(F.LT.1D-6*SU)GO TO 40
   10 SU=SU+F
   40 SZ=.5D0*SU
      IF(J.LT.ICNT(1))RETURN
      ICNT(1)=ICNT(1)+5
      ITGL=.TRUE.
      RETURN
  100 SU=DV(3)
      J=0
      IF(SU.GT.0.9999D0)GO TO 120
      C1=4D0/PI
      C2=PI*PI*TZ
      C3=.5D0*PI*DV(3)
      C4=PI*DV(2)
      C5=PI*DV(4)
      DO 110 J=1,200
      F=C1*DDEXP(-J*J*C2)/J
      IF(F.LT.1D-6*DABS(SU))GO TO 120
  110 SU=SU+F*DSIN(J*C3)*DCOS(J*C4)*DCOS(J*C5)
  120 SZ=SU
      IF(J.LT.ICNT(2))RETURN
      ICNT(2)=ICNT(2)+5
      ITGL=.FALSE.
      RETURN
      END
      SUBROUTINE ZSCC(ITGL,ICNT)
      INTEGER ICNT(2)
      LOGICAL ITGL
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/P1VARS/DV(5),SR,SZ,TR,TZ,PI
      IF(ITGL)GO TO 100
      C=.5D0*DV(3)
      CB=.5D0/DSQRT(TZ)
      CC=2D0*CB
      C1=CB*(C+DV(4)-DV(2))
      C2=CB*(C-DV(4)+DV(2))
      C3=CB*(C+DV(4)+DV(2))
      C4=CB*(C-DV(4)-DV(2))
      SU=DERF(C1)+DERF(C2)-DERF(C3)-DERF(C4)
      N=(C3+10D0)/CC
      IF(N.LE.1)GO TO 40
      DO 10 J=1,100
      F=DERF(C1-J*CC)+DERF(C2-J*CC)-DERF(C3-J*CC)-DERF(C4-J*CC)
      F=F+DERF(C1+J*CC)+DERF(C2+J*CC)-DERF(C3+J*CC)-DERF(C4+J*CC)
      IF(F.LT.1D-6*SU)GO TO 40
```

```
 10 SU=SU+F
 40 SZ=.5D0*SU
    IF(J.LT.ICNT(1))RETURN
    ICNT(1)=ICNT(1)+5
    ITGL=.TRUE.
    RETURN
100 SU=0D0
    C1=4D0/PI
    C2=PI*PI*TZ
    C3=.5D0*PI*DV(3)
    C4=PI*DV(2)
    C5=PI*DV(4)
    DO 110 J=1,200
    F=C1*DDEXP(-J*J*C2)/J
    IF(F.LT.1D-6*DABS(SU))GO TO 120
110 SU=SU+F*DSIN(J*C3)*DSIN(J*C4)*DSIN(J*C5)
120 SZ=SU
    IF(J.LT.ICNT(2))RETURN
    ICNT(2)=ICNT(2)+5
    ITGL=.FALSE.
    RETURN
    END
    SUBROUTINE ZSNC(ITGL,ICNT)
    INTEGER ICNT(2)
    LOGICAL ITGL
    IMPLICIT REAL*8 (A-H,O-Z)
    COMMON/P1VARS/DV(5),SR,SZ,TR,TZ,PI
    IF(ITGL)GO TO 100
    C=.5D0*DV(3)
    CB=.5D0/DSQRT((TZ))
    CC=2D0*CB
    C1=CB*(C+DV(4)-DV(2))
    C2=CB*(C-DV(4)+DV(2))
    C3=CB*(C+DV(4)+DV(2))
    C4=CB*(C-DV(4)-DV(2))
    SU=DERF(C1)+DERF(C2)+DERF(C3)+DERF(C4)
    N=(C3+10D0)/CC
    IF(N.LE.1)GO TO 40
    SIGN=-1D0
    DO 10 J=1,100
    F=DERF(C1-J*CC)+DERF(C2-J*CC)+DERF(C3-J*CC)+DERF(C4-J*CC)
    F=F+DERF(C1+J*CC)+DERF(C2+J*CC)+DERF(C3+J*CC)+DERF(C4+J*CC)
    IF(F.LT.1D-6*SU)GO TO 40
    SU=SU+SIGN*F
    IF(DABS(SU).LT.1D-6)GO TO 40
 10 SIGN=-SIGN
 40 SZ=.5D0*SU
    IF(J.LE.ICNT(1))RETURN
    ICNT(1)=ICNT(1)+5
    ITGL=.TRUE.
```

```
      RETURN
  100 SU=0D0
      C1=6D0/PI
      C2=.25D0*PI*PI*TZ
      C3=.25D0*PI*DV(3)
      C4=PI*DV(2)
      C5=PI*DV(4)
      DO 110 J=1,200
      J1=2*J+1
      F=C1*DDEXP(-J1*J1*C2)/J1
      IF(F.LT.1D-6*DABS(SU))GO TO 120
      SU=SU+F*DSIN(J1*C3)*DCOS(J1*C4)*DCOS(J1*C5)
      IF(DABS(SU).LT.1D-6)GO TO 120
  110 CONTINUE
  120 SZ=SU
      IF(J.LT.ICNT(2))RETURN
      ICNT(2)=ICNT(2)+5
      ITGL=.FALSE.
      RETURN
      END
      DOUBLE PRECISION FUNCTION DERF(Z)
      DOUBLE PRECISION Z
      DOUBLE PRECISION C(5),P,T,T1,T2,SIGN
      DATA C/.254829592D0,-.284496736D0,1.421413741D0,-1.453152027D0,
     &1.061405429D0/
      DATA P/.3275911D0/
      SIGN=1D0
      IF(Z.GT.0D0) GO TO 10
      SIGN=-1D0
   10 IF(DABS(Z).GT.9)GO TO 20
      T=1D0/(1D0+P*DABS(Z))
      T1=((((C(5)*T+C(4))*T+C(3))*T+C(2))*T+C(1))*T
      T2=-Z*Z
      DERF=SIGN*(1D0-T1*DEXP(T2))
      RETURN
   20 DERF=1D0*SIGN
      RETURN
      END
```

```
C   THIS IS THESIS PROGRAM 4 - THE BOUNDED CASES WITH LOG TIMESTEPS
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION TD(200),PD(200),ITP(10),ITQ(10),IPP(10),IPQ(10)
      DIMENSION DV(11),UD(3),UWD(3),UFD(3),URD(2),F(11),TFAC(3),S(3)
      DIMENSION ITYP(3),ICNTA(3,2),ICNTB(3,2),ICNTC(3,2),ICNTD(3,2)
      DIMENSION ICNTE(3,2),ICNTF(3,2),DT(11),PDB(200)
      LOGICAL ITGLA(3),ITGLB(3),ITGLC(3),ITGLD(3),ITGLE(3),ITGLF(3)
      LOGICAL SFLAG
      COMMON/P2VARS/ID,UD,UWD,UFD,TFAC,S,T,PI
      PI=4D0*DATAN(1D0)
      DT(1)=1D0
      DDT=10D0**1D-2
      DO 5 I=2,11
    5 DT(I)=DT(I-1)*DDT
   10 PRINT,'THERE ARE SIX TYPES OF GEOMETRIES AVAILABLE FOR EACH'
      PRINT,'DIMENSION; PLEASE CHOOSE ANY ONE FOR EACH, SEPARATED BY'
      PRINT,'COMMAS, LIKE     3,4,3  :'
      PRINT,'   1. PLANE, NOFLOW, NOFLOW'
      PRINT,'   2. PLANE, CONSTP, CONSTP'
      PRINT,'   3. PLANE, NOFLOW, CONSTP'
      PRINT,'   4.  SLAB, NOFLOW, NOFLOW'
      PRINT,'   5.  SLAB, CONSTP, CONSTP'
      PRINT,'   6.  SLAB, NOFLOW, CONSTP'
      PRINT,'----------------WHICH TYPES FOR X,Y,Z?'
      READ,ITYP(1),ITYP(2),ITYP(3)
      PRINT,'IS THIS DRAWDOWN (0) OR BUILDUP (1) ?'
      READ,IBLDP
   20 PRINT,'THE DIMENSIONLESS VARIABLES ARE:'
      PRINT,'   1. XD   2. YD   3. ZD'
      PRINT,'   4. XWD  5. YWD  6. ZWD'
      PRINT,'   7. XFD  8. YFD  9. ZFD'
      PRINT,'    10. YE/XE 11. ZE/XE'
      PRINT,'YOU MAY MULTIPLY OR DIVIDE THE DEFAULT VALUES FOR PD AND'
      PRINT,'TD BY ANY OF THESE TO CREATE THE TD AND PD OF YOUR CHOICE.'
      PRINT,'TD DEFAULTS TO:'
      PRINT,'                    K * T'
      PRINT,'TD = TDA = ----------------------'
      PRINT,'           PHI * MU * CT * XE * YE'
      PRINT,'AND PD DEFAULTS TO:'
      PRINT,'     2 * PI * K * ZE * DELTA-P'
      PRINT,'PD = ------------------------'
      PRINT,'            Q * MU'
      NTP=0
   21 PRINT,'WHAT TO MULTIPLY TD BY (0 TO QUIT)?'
      READ,I
      IF(I.EQ.0)GO TO 22
      NTP=NTP+1
      ITP(NTP)=I
      GO TO 21
   22 NTQ=0
```

```
 23 PRINT,'WHAT TO DIVIDE TO BY (0 TO QUIT)?'
    READ,I
    IF(I.EQ.0)GO TO 24
    NTQ=NTQ+1
    ITQ(NTQ)=I
    GO TO 23
 24 NPP=0
 25 PRINT,'WHAT TO MULTIPLY PD BY (0 TO QUIT)?'
    READ,I
    IF(I.EQ.0)GO TO 26
    NPP=NPP+1
    IPP(NPP)=I
    GO TO 25
 26 NPQ=0
 27 PRINT,'WHAT TO DIVIDE PD BY (0 TO QUIT)?'
    READ,I
    IF(I.EQ.0)GO TO 30
    NPQ=NPQ+1
    IPQ(NPQ)=I
    GO TO 27
 30 PRINT,'WHEN ENTERING VALUES FOR THE DIMENSIONLESS VARIABLES TO'
    PRINT,'FOLLOW, ENTER 0 FOR THOSE THAT DO NOT APPLY TO THE PROBLEM:'
    PRINT,'WHAT ARE XD,YD,ZD?'
    READ,UD(1),UD(2),UD(3)
    PRINT,'WHAT ARE XWD,YWD,ZWD?'
    READ,UWD(1),UWD(2),UWD(3)
    PRINT,'WHAT ARE XFD,YFD,ZFD?'
    READ,UFD(1),UFD(2),UFD(3)
    PRINT,'WHAT ARE YE/XE, ZE/XE?'
    READ,URD(1),URD(2)
    DO 996 I=1,3
    DV(I)=UD(I)
    DV(I+3)=UWD(I)
996 DV(I+6)=UFD(I)
    DV(10)=URD(1)
    DV(11)=URD(2)
    FT=1D0
    IF(NTP.EQ.0)GO TO 32
    DO 31 I=1,NTP
 31 FT=FT/DV(ITP(I))
 32 IF(NTQ.EQ.0)GO TO 34
    DO 33 I=1,NTQ
 33 FT=FT*DV(ITQ(I))
 34 FP=1D0
    IF(NPP.EQ.0)GO TO 36
    DO 35 I=1,NPP
 35 FP=FP*DV(IPP(I))
 36 IF(NPQ.EQ.0)GO TO 38
    DO 37 I=1,NPQ
 37 FP=FP/DV(IPQ(I))
```

```
38 CONTINUE
   TFAC(1)=URD(1)
   TFAC(2)=1D0/URD(1)
   TFAC(3)=URD(1)/URD(2)/URD(2)
39 IF(IBLDP.EQ.0)GO TO 40
   PRINT,'WHAT DIMENSIONLESS TIME (YOUR UNITS) IS SHUT-IN TIME?'
   READ,TDP
40 PRINT,'FOR THE TIME RANGE, PLEASE USE LOG-CYCLE NOTATION:'
   PRINT,'FOR A RANGE OF TD=10**-3 TO 10**2, ENTER -3,2.'
   PRINT,'WHAT (BEGIN,END) CYCLES?'
   READ,NCI,NCF
   PRINT,'LOWER AND UPPER LIMITS ON PD (YOUR UNITS)?'
   READ,PDLOW,PDHI
   PRINT,'**** MODEL VERIFICATION ****'
   PRINT,'X-AXIS GEOMETRY IS TYPE',ITYP(1)
   PRINT,'Y-AXIS GEOMETRY IS TYPE',ITYP(2)
   PRINT,'Z-AXIS GEOMETRY IS TYPE',ITYP(3)
   IF(NTP+NTQ.GT.0)PRINT,'---TD HAS BEEN MODIFIED.'
   IF(NPP+NPQ.GT.0)PRINT,'---PD HAS BEEN MODIFIED.'
   PRINT,'XD,YD,ZD     ='
   PRINT 999,DV(1),DV(2),DV(3)
   PRINT,'XWD,YWD,ZWD  ='
   PRINT 999,DV(4),DV(5),DV(6)
   PRINT,'XFD,YFD,ZFD  ='
   PRINT 999,DV(7),DV(8),DV(9)
   PRINT,'YE/XE,ZE/XE  ='
   PRINT 999,DV(10),DV(11)
999 FORMAT(1X,3D12.5)
   PRINT,'START,END CYCLES ARE SET TO :',NCI,NCF
   PRINT,'LOWER LIMIT OF PD =',PDLOW
   PRINT,'UPPER LIMIT OF PD =',PDHI
   PRINT,'========== IS EVERYTHING OKAY (1=YES)?'
   READ,I
   IF(I.NE.1)GO TO 60
   DO 41 I=1,3
   ICNTA(I,1)=5
   ICNTA(I,2)=5
   ICNTB(I,1)=5
   ICNTB(I,2)=5
   ICNTC(I,1)=5
   ICNTC(I,2)=5
   ICNTD(I,1)=5
   ICNTD(I,2)=5
   ICNTE(I,1)=5
   ICNTE(I,2)=5
   ICNTF(I,1)=5
   ICNTF(I,2)=5
   ITGLA(I)=.FALSE.
   ITGLB(I)=.FALSE.
   ITGLC(I)=.FALSE.
```

```
      ITGLD(I)=.FALSE.
      ITGLE(I)=.FALSE.
   41 ITGLF(I)=.FALSE.
      FA=PI*FP*FT
      DO 997 I=1,3
  997 IF(ITYP(I).GT.3)FA=FA/UFD(I)
      NCI=NCI-2
      NC=NCF-NCI
      TD(1)=0D0
      PD(1)=0D0
      I=1
      SFLAG=.FALSE.
      TDU=10D0**NCI
      F(1)=0D0
      DO 53 NCC=1,NC
      PRINT,'NOW CALCULATING FOR LOG CYCLE',NCI+NCC-1
      TDL=TDU
      DO 52 NL=1,10
      DO 51 J=2,11
      T=FT*TDL*DT(J)
      DO 998 K=1,3
      ID=K
      IF(ITYP(K).EQ.1)CALL SPNN(ITGLA,ICNTA)
      IF(ITYP(K).EQ.2)CALL SPCC(ITGLB,ICNTB)
      IF(ITYP(K).EQ.3)CALL SPNC(ITGLC,ICNTC)
      IF(ITYP(K).EQ.4)CALL SSNN(ITGLD,ICNTD)
      IF(ITYP(K).EQ.5)CALL SSCC(ITGLE,ICNTE)
      IF(ITYP(K).EQ.6)CALL SSNC(ITGLF,ICNTF)
  998 CONTINUE
   51 F(J)=S(1)*S(2)*S(3)
      IF(SFLAG)GO TO 919
      PD(1)=F(2)*FA*TDL
      SFLAG=.TRUE.
  919 I=I+1
      TD(I)=TDL*DT(11)
      PD(I)=PD(I-1)
      DO 991 J=2,11
  991 PD(I)=PD(I)+(F(J)+F(J-1))*FA*TDL*(DT(J)-DT(J-1))
      IF((PD(I).GE.PDLOW).AND.(NCC.GT.2))GO TO 400
      I=I-1
      PD(I)=PD(I+1)
      TD(I)=TD(I+1)
      GO TO 401
  400 IF(PD(I).LE.PDHI)GO TO 401
      I=I-1
      GO TO 600
  401 F(1)=F(11)
   52 TDL=TDL*DT(11)
   53 TDU=TDU*10D0
  600 IF(IBLDP.EQ.0)GO TO 54
```

```
      PDB(1)=0D0
      K=0
      DO 800 J=1,3
      IF((ITYP(J).EQ.1).OR.(ITYP(J).EQ.4))GO TO 800
      K=1
  800 CONTINUE
      PDC=2D0*PI*TDP
      IF(K.EQ.1)PDC=0D0
      J=1
  801 J=J+1
      T=TDP+TD(J)
      IL=1
      IH=I
  802 K=(IL+IH)/2
      IF(TD(K).GT.T)GO TO 803
      IL=K
      IF((IL+1).GE.IH)GO TO 804
      GO TO 802
  803 IH=K
      IF((IH-1).LE.IL)GO TO 804
      GO TO 802
  804 PDI=PD(IL)+(T-TD(IL))*(PD(IH)-PD(IL))/(TD(IH)-TD(IL))
      PDB(J)=PDI-PD(J)-PDC
      IF(T.LT.TD(I))GO TO 801
      PRINT,'NOW WRITING TD AND PDB TO TEMP FILE 13.'
      K=J-1
      WRITE(13,1000)K
      WRITE(13,1100)(TD(KK),PDB(KK),KK=2,J)
      GO TO 55
   54 PRINT,'NOW WRITING TD AND PD TO TEMP FILE 13.'
      NP=I-1
      WRITE(13,1000)NP
      WRITE(13,1100)(TD(J),PD(J),J=2,I)
 1000 FORMAT(I3)
 1100 FORMAT(2E12.4)
   55 DO 1110 ID=1,3
      IF(ITYP(ID).EQ.1)PRINT,ICNTA(ID,1),ICNTA(ID,2)
      IF(ITYP(ID).EQ.2)PRINT,ICNTB(ID,1),ICNTB(ID,2)
      IF(ITYP(ID).EQ.3)PRINT,ICNTC(ID,1),ICNTC(ID,2)
      IF(ITYP(ID).EQ.4)PRINT,ICNTD(ID,1),ICNTD(ID,2)
      IF(ITYP(ID).EQ.5)PRINT,ICNTE(ID,1),ICNTE(ID,2)
 1110 IF(ITYP(ID).EQ.6)PRINT,ICNTF(ID,1),ICNTF(ID,2)
      IF(IBLDP.EQ.1)I=J
      PRINT,'DO YOU WISH TO SEE A TABLE (1=YES)?'
      READ,J
      IF(J.NE.1)GO TO 60
      NR=(I+2)/3
      I1=NR
      I2=2*NR
      PRINT,'    TD        PD        TD        PD        TD        PD'
```

```
      PRINT,'=============================================================='
      IF(IBLDP.EQ.1)GO TO 700
      DO 56 J=1,NR
   56 WRITE(6,1200)TD(J),PD(J),TD(J+I1),PD(J+I1),TD(J+I2),PD(J+I2)
      GO TO 60
 1200 FORMAT(6E10.3)
  700 DO 59 J=1,NR
   59 WRITE(6,1200)TD(J),PDB(J),TD(J+I1),PDB(J+I1),TD(J+I2),
     &PDB(J+I2)
   60 PRINT,'OPTIONS:'
      PRINT,'  1. NEW CYCLE RANGE FOR TIME.'
      PRINT,'  2. NEW VALUES FOR DIMENSIONLESS VARIABLES.'
      PRINT,'  3. NEW DEFINITIONS FOR TD AND/OR PD.'
      PRINT,'  4. NEW MODEL GEOMETRY.'
      PRINT,'  5. QUIT.'
      PRINT,'======== WHICH ONE?'
      READ,I
      IF(I.LT.1)GO TO 60
      GO TO (40,30,20,10,70),I
   70 STOP
      END
      SUBROUTINE SPNN(ITGL,ICNT)
      INTEGER ICNT(3,2)
      LOGICAL ITGL(3)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/P2VARS/ID,UD(3),UWD(3),UFD(3),TFAC(3),S(3),T,PI
      IF(ITGL(ID))GO TO 100
      C1=UD(ID)-UWD(ID)
      C2=UD(ID)+UWD(ID)
      C3=.25D0/(T*TFAC(ID))
      SU=DDEXP(-C1*C1*C3)+DDEXP(-C2*C2*C3)
      DO 10 J=1,100
      F=DDEXP(-(C1-2*J)**2*C3)+DDEXP(-(C2-2*J)**2*C3)
      F=F+DDEXP(-(C1+2*J)**2*C3)+DDEXP(-(C2+2*J)**2*C3)
      IF(F.LT.1D-6*SU)GO TO 40
   10 SU=SU+F
   40 S(ID)=SU/2D0/DSQRT(PI*T*TFAC(ID))
      IF(J.LT.ICNT(ID,1))RETURN
      ITGL(ID)=.TRUE.
      ICNT(ID,1)=ICNT(ID,1)+5
      RETURN
  100 SU=1D0
      C1=PI*PI*T*TFAC(ID)
      C2=PI*UWD(ID)
      C3=PI*UD(ID)
      DO 110 J=1,200
      F=2D0*DDEXP(-J*J*C1)
      IF(F.LT.1D-6*DABS(SU))GO TO 120
  110 SU=SU+F*DCOS(J*C2)*DCOS(J*C3)
  120 S(ID)=SU
```

```
      IF(J.LT.ICNT(ID,2))RETURN
      ITGL(ID)=.FALSE.
      ICNT(ID,2)=ICNT(ID,2)+5
      RETURN
      END
      SUBROUTINE SPCC(ITGL,ICNT)
      INTEGER ICNT(3,2)
      LOGICAL ITGL(3)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/P2VARS/ID,UD(3),UWD(3),UFD(3),TFAC(3),S(3),T,PI
      IF((T*TFAC(ID)).LT.(5D0/PI/PI))GO TO 5
      S(ID)=0D0
      RETURN
    5 IF(ITGL(ID))GO TO 100
      C1=UD(ID)-UWD(ID)
      C2=UD(ID)+UWD(ID)
      C3=.25D0/(T*TFAC(ID))
      SU=DDEXP(-C1*C1*C3)-DDEXP(-C2*C2*C3)
      DO 10 J=1,100
      F=DDEXP(-(C1-2*J)**2*C3)-DDEXP(-(C2-2*J)**2*C3)
      F=F+DDEXP(-(C1+2*J)**2*C3)-DDEXP(-(C2+2*J)**2*C3)
      IF(F.LT.1D-6*SU)GO TO 40
   10 SU=SU+F
   40 S(ID)=SU/2D0/DSQRT(PI*T*TFAC(ID))
      IF(J.LT.ICNT(ID,1))RETURN
      ICNT(ID,1)=ICNT(ID,1)+5
      ITGL(ID)=.TRUE.
      RETURN
  100 SU=0D0
      C1=PI*PI*T*TFAC(ID)
      C2=PI*UWD(ID)
      C3=PI*UD(ID)
      DO 110 J=1,200
      F=2D0*DDEXP(-J*J*C1)
      IF(F.LT.1D-6*DABS(SU))GO TO 120
  110 SU=SU+F*DSIN(J*C2)*DSIN(J*C3)
  120 S(ID)=SU
      IF(J.LT.ICNT(ID,2))RETURN
      ICNT(ID,2)=ICNT(ID,2)+5
      ITGL(ID)=.FALSE.
      RETURN
      END
      SUBROUTINE SPNC(ITGL,ICNT)
      INTEGER ICNT(3,2)
      LOGICAL ITGL(3)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/P2VARS/ID,UD(3),UWD(3),UFD(3),TFAC(3),S(3),T,PI
      IF((T*TFAC(ID)).LT.(2D1/PI/PI))GO TO 5
      S(ID)=0D0
      RETURN
```

```
  5 IF(ITGL(ID))GO TO 100
    C1=UD(ID)-UWD(ID)
    C2=UD(ID)+UWD(ID)
    C3=.25D0/(T*TFAC(ID))
    SIGN=-1D0
    SU=DDEXP(-C1*C1*C3)+DDEXP(-C2*C2*C3)
    DO 10 J=1,100
    F=DDEXP(-(C1-2*J)**2*C3)+DDEXP(-(C2-2*J)**2*C3)
    F=F+DDEXP(-(C1+2*J)**2*C3)+DDEXP(-(C2+2*J)**2*C3)
    IF(F.LT.1D-6*SU)GO TO 40
    SU=SU+SIGN*F
    IF(DABS(SU).LT.1D-6)GO TO 40
 10 SIGN=-SIGN
 40 S(ID)=SU/2D0/DSQRT(PI*T*TFAC(ID))
    IF(J.LT.ICNT(ID,1))RETURN
    ICNT(ID,1)=ICNT(ID,1)+5
    ITGL(ID)=.TRUE.
    RETURN
100 SU=0D0
    C1=PI*PI*T*TFAC(ID)*.25D0
    C2=PI*UWD(ID)
    C3=PI*UD(ID)
    DO 110 J=1,200
    J1=2*J+1
    F=2D0*DDEXP(-J1*J1*C1)
    IF(F.LT.1D-6*DABS(SU))GO TO 120
    SU=SU+F*DCOS(J1*C2)*DCOS(J1*C3)
    IF(DABS(SU).LT.1D-6)GO TO 120
110 CONTINUE
120 S(ID)=SU
    IF(J.LT.ICNT(ID,2))RETURN
    ICNT(ID,2)=ICNT(ID,2)+5
    ITGL(ID)=.FALSE.
    RETURN
    END
    SUBROUTINE SSNN(ITGL,ICNT)
    INTEGER ICNT(3,2)
    LOGICAL ITGL(3)
    IMPLICIT REAL*8 (A-H,O-Z)
    COMMON/P2VARS/ID,UD(3),UWD(3),UFD(3),TFAC(3),S(3),T,PI
    DEL=DABS(UD(ID)-UWD(ID))
    IF(DEL.GT.(UFD(ID)*.5D0))GO TO 6
    DEL=UFD(ID)*.5D0-DEL
    IF((T*TFAC(ID)).GT.(DEL*DEL*.05D0))GO TO 9
    S(ID)=1D0
    RETURN
  6 DEL=DEL-UFD(ID)*.5D0
    IF((T*TFAC(ID)).GT.(DEL*DEL*.05D0))GO TO 9
    S(ID)=0D0
    RETURN
```

```
  9 IF(ITGL(ID))GO TO 100
    IF(UFD(ID).LT.0.99D0)GO TO 5
    SU=2D0*UFD(ID)
    GO TO 40
  5 C=.5D0*UFD(ID)
    CB=.5D0/DSQRT(T*TFAC(ID))
    CC=2D0*CB
    C1=CB*(C+UD(ID)-UWD(ID))
    C2=CB*(C-UD(ID)+UWD(ID))
    C3=CB*(C+UD(ID)+UWD(ID))
    C4=CB*(C-UD(ID)-UWD(ID))
    SU=DERF(C1)+DERF(C2)+DERF(C3)+DERF(C4)
    N=(C3+10D0)/CC
    IF(N.LE.1)GO TO 40
    DO 10 J=1,100
    F=DERF(C1-J*CC)+DERF(C2-J*CC)+DERF(C3-J*CC)+DERF(C4-J*CC)
    F=F+DERF(C1+J*CC)+DERF(C2+J*CC)+DERF(C3+J*CC)+DERF(C4+J*CC)
    IF(F.LT.1D-6*SU)GO TO 40
 10 SU=SU+F
 40 S(ID)=.5D0*SU
    IF(J.LT.ICNT(ID,1))RETURN
    ICNT(ID,1)=ICNT(ID,1)+5
    ITGL(ID)=.TRUE.
    RETURN
100 SU=UFD(ID)
    J=0
    IF(SU.GT.0.9999D0)GO TO 120
    C1=4D0/PI
    C2=PI*PI*T*TFAC(ID)
    C3=.5D0*PI*UFD(ID)
    C4=PI*UWD(ID)
    C5=PI*UD(ID)
    DO 110 J=1,200
    F=C1*DDEXP(-J*J*C2)/J
    IF(F.LT.1D-6*DABS(SU))GO TO 120
110 SU=SU+F*DSIN(J*C3)*DCOS(J*C4)*DCOS(J*C5)
120 S(ID)=SU
    IF(J.LT.ICNT(ID,2))RETURN
    ICNT(ID,2)=ICNT(ID,2)+5
    ITGL(ID)=.FALSE.
    RETURN
    END
    SUBROUTINE SSCC(ITGL,ICNT)
    INTEGER ICNT(3,2)
    LOGICAL ITGL(3)
    IMPLICIT REAL*8 (A-H,O-Z)
    COMMON/P2VARS/ID,UD(3),UWD(3),UFD(3),TFAC(3),S(3),T,PI
    IF((T*TFAC(ID)).LT.(5D0/PI/PI))GO TO 5
    S(ID)=0D0
    RETURN
```

```
  5 DEL=DABS(UD(ID)-UWD(ID))
    IF(DEL.GT.(UFD(ID)*.5D0))GO TO 6
    DEL=UFD(ID)*.5D0-DEL
    IF((T*TFAC(ID)).GT.(DEL*DEL*.05D0))GO TO 9
    S(ID)=1D0
    RETURN
  6 DEL=DEL-UFD(ID)*.5D0
    IF((T*TFAC(ID)).GT.(DEL*DEL*.05D0))GO TO 9
    S(ID)=0D0
    RETURN
  9 IF(ITGL(ID))GO TO 100
    C=.5D0*UFD(ID)
    CB=.5D0/DSQRT(T*TFAC(ID))
    CC=2D0*CB
    C1=CB*(C+UD(ID)-UWD(ID))
    C2=CB*(C-UD(ID)+UWD(ID))
    C3=CB*(C+UD(ID)+UWD(ID))
    C4=CB*(C-UD(ID)-UWD(ID))
    SU=DERF(C1)+DERF(C2)-DERF(C3)-DERF(C4)
    N=(C3+10D0)/CC
    IF(N.LE.1)GO TO 40
    DO 10 J=1,100
    F=DERF(C1-J*CC)+DERF(C2-J*CC)-DERF(C3-J*CC)-DERF(C4-J*CC)
    F=F+DERF(C1+J*CC)+DERF(C2+J*CC)-DERF(C3+J*CC)-DERF(C4+J*CC)
    IF(F.LT.1D-6*SU)GO TO 40
 10 SU=SU+F
 40 S(ID)=.5D0*SU
    IF(J.LT.ICNT(ID,1))RETURN
    ICNT(ID,1)=ICNT(ID,1)+5
    ITGL(ID)=.TRUE.
    RETURN
100 SU=0D0
    C1=4D0/PI
    C2=PI*PI*T*TFAC(ID)
    C3=.5D0*PI*UFD(ID)
    C4=PI*UWD(ID)
    C5=PI*UD(ID)
    DO 110 J=1,200
    F=C1*DDEXP(-J*J*C2)/J
    IF(F.LT.1D-6*DABS(SU))GO TO 120
110 SU=SU+F*DSIN(J*C3)*DSIN(J*C4)*DSIN(J*C5)
120 S(ID)=SU
    IF(J.LT.ICNT(ID,2))RETURN
    ICNT(ID,2)=ICNT(ID,2)+5
    ITGL(ID)=.FALSE.
    RETURN
    END
    SUBROUTINE SSNC(ITGL,ICNT)
    INTEGER ICNT(3,2)
    LOGICAL ITGL(3)
```

```
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/P2VARS/ID,UD(3),UWD(3),UFD(3),TFAC(3),S(3),T,PI
      IF((T*TFAC(ID)).LT.(2D1/PI/PI))GO TO 5
      S(ID)=0D0
      RETURN
    5 DEL=DABS(UD(ID)-UWD(ID))
      IF(DEL.GT.(UFD(ID)*.5D0))GO TO 6
      DEL=UFD(ID)*.5D0-DEL
      IF((T*TFAC(ID)).GT.(DEL*DEL*.05D0))GO TO 9
      S(ID)=1D0
      RETURN
    6 DEL=DEL-UFD(ID)*.5D0
      IF((T*TFAC(ID)).GT.(DEL*DEL*.05D0))GO TO 9
      S(ID)=0D0
      RETURN
    9 IF(ITGL(ID))GO TO 100
      C=.5D0*UFD(ID)
      CB=.5D0/DSQRT((T*TFAC(ID)))
      CC=2D0*CB
      C1=CB*(C+UD(ID)-UWD(ID))
      C2=CB*(C-UD(ID)+UWD(ID))
      C3=CB*(C+UD(ID)+UWD(ID))
      C4=CB*(C-UD(ID)-UWD(ID))
      SU=DERF(C1)+DERF(C2)+DERF(C3)+DERF(C4)
      N=(C3+10D0)/CC
      IF(N.LE.1)GO TO 40
      SIGN=-1D0
      DO 10 J=1,100
      F=DERF(C1-J*CC)+DERF(C2-J*CC)+DERF(C3-J*CC)+DERF(C4-J*CC)
      F=F+DERF(C1+J*CC)+DERF(C2+J*CC)+DERF(C3+J*CC)+DERF(C4+J*CC)
      IF(F.LT.1D-6*SU)GO TO 40
      SU=SU+SIGN*F
      IF(DABS(SU).LT.1D-6)GO TO 40
   10 SIGN=-SIGN
   40 S(ID)=.5D0*SU
      IF(J.LE.ICNT(ID,1))RETURN
      ICNT(ID,1)=ICNT(ID,1)+5
      ITGL(ID)=.TRUE.
      RETURN
  100 SU=0D0
      C1=8D0/PI
      C2=.25D0*PI*PI*T*TFAC(ID)
      C3=.25D0*PI*UFD(ID)
      C4=.5D0*PI*UWD(ID)
      C5=.5D0*PI*UD(ID)
      DO 110 J=1,200
      J1=2*J-1
      F=C1*DDEXP(-J1*J1*C2)/J1
      IF(F.LT.1D-6*DABS(SU))GO TO 120
      SU=SU+F*DSIN(J1*C3)*DCOS(J1*C4)*DCOS(J1*C5)
```

```
      IF(DABS(SU).LT.1D-6)GO TO 120
  110 CONTINUE
  120 S(ID)=SU
      IF(J.LT.ICNT(ID,2))RETURN
      ICNT(ID,2)=ICNT(ID,2)+5
      ITGL(ID)=.FALSE.
      RETURN
      END
      FUNCTION DDEXP(ARG)
      IMPLICIT REAL*8 (A-Z)
      IF(DABS(ARG).LE.87D0)GO TO 20
      IF(ARG.GT.87D0)GO TO 10
      DDEXP=0D0
      RETURN
   10 DDEXP=1D38
      RETURN
   20 DDEXP=DEXP(ARG)
      RETURN
      END
      DOUBLE PRECISION FUNCTION DERF(Z)
      DOUBLE PRECISION Z
      DOUBLE PRECISION C(5),P,T,T1,T2,SIGN
      DATA C/.254829592D0,-.284496736D0,1.421413741D0,-1.453152027D0,
     &1.061405429D0/
      DATA P/.3275911D0/
      SIGN=1D0
      IF(Z.GT.0D0) GO TO 10
      SIGN=-1D0
   10 IF(DABS(Z).GT.9)GO TO 20
      T=1D0/(1D0+P*DABS(Z))
      T1=((((C(5)*T+C(4))*T+C(3))*T+C(2))*T+C(1))*T
      T2=-Z*Z
      DERF=SIGN*(1D0-T1*DEXP(T2))
      RETURN
   20 DERF=1D0*SIGN
      RETURN
      END
```

BIBLIOGRAPHY

Buhidma, I.M. and R. Raghavan, "Transient Pressure Behavior of Partially Penetrating Wells Subject to Bottomwater Drive", Jrnl. Pet. Tech., July 1980, p. 1251.

Carslaw, H.S., and J.C. Jaeger, Conduction of Heat in Solids, Oxford University Press, first edition, 1946.

Cinco-L., H., W.C. Brigham, M.J. Economides, F.G. Miller, H.J. Ramey, Jr., A. Barelli, and G. Manetti, "A Parallelepiped Model to Analyze the Pressure Behavior of Geothermal Steam Wells Penetrating Vertical Fractures", Paper SPE 8231, presented at 54th Annual Fall Technical Conference and Exhibition of the SPE of AIME, Sept. 1979.

Cinco-L., H., H.J. Ramey, Jr. and F.G. Miller, "Unsteady-State Pressure Distribution Created by a Well With an Inclined Fracture", Paper SPE 5591, presented at 50th Annual Fall Technical Conference and Exhibition of the SPE of AIME, Sept-Oct. 1975.

Cinco-L., H., F. Samaniego-V., N. Dominguez-A., "Transient Pressure Behavior for a Well With a Finite-Conductivity Vertical Fracture", Soc. Pet. Eng. Jrnl., Aug. 1978, p. 253.

deSwann-O., A., "Analytic Solutions for Determining Naturally Fractured Reservoir Properties by Well Testing", Soc. Pet. Eng. Jrnl., June 1976, p. 117.

Economides, M.J., D. Ogbe, F.G. Miller, H. Cinco-L., and E.L. Fehlberg, "Pressure Buildup Analysis of Geothermal Steam Wells Using a Parallelepiped Model", Paper SPE 8886, presented at 50th Annual California Regional Meeting of the SPE of AIME, April 1980.

Gringarten, A.C., "Reservoir Limit Testing for Fractured Wells", Paper SPE 7452, presented at 53rd Annual Fall Technical Conference and Exhibition of the SPE of AIME, Oct. 1978.

Gringarten, A.C. and H.J. Ramey, Jr., "The Use of Source and Green's Functions in Solving Unsteady-Flow Problems in Reservoirs", Soc. Pet. Eng. Jrnl., Oct. 1973, p. 285.

Gringarten, A.C. and H.J. Ramey, Jr., "Unsteady-State Pressure Distributions Created by a Well With a Single Horizontal Fracture, Partial Penetration, or Restricted Entry", Soc. Pet. Eng. Jrnl., Aug. 1974, p. 413.

Gringarten, A.C., H.J. Ramey, Jr., and R. Raghavan, "Unsteady-State Pressure Distributions Created by a Well With a Single Infinite-Conductivity Vertical Fracture", Soc. Pet. Eng. Jrnl., Aug. 1974, p. 347.

Kohlhaas, C.A., C. del Giudice, and W.A. Abbott, "Applications of Linear & Spherical Flow Analysis Techniques to Field Problems-Case Studies", Paper SPE 11088, presented at 57th Annual Fall Technical Conference and Exhibition of the SPE of AIME, Sept. 1982.

Lord Kelvin, Mathematical and Physical Papers, Cambridge at the University Press, 1884, Vol. 2, p. 41.

Nisle, R.G., "The Effect of Partial Penetration on Pressure Buildup in Oil Wells", Trans. AIME, Vol. 213, 1958, p. 85.

Prats, M., "Effect of Vertical Fractures on Reservoir Behavior - Incompressible Fluid Case", Soc. Pet. Eng. Jrnl., June 1961, p. 105.

Prats, M., P. Hazebroek, and W.R. Strickler, "Effect of Vertical Fractures on Reservoir Behavior - Compressible Fluid Case", Soc. Pet. Eng. Jrnl., June 1962, p. 87.

Raghavan, R. and N. Hadinoto, "Analysis of Pressure Data for Fractured Wells: The Constant-Pressure Outer Boundary", Soc. Pet. Eng. Jrnl., Apr. 1978, p. 139.

Raghavan, R., A. Uraiet and G.W. Thomas, "Vertical Fracture Height: Effect on Transient flow Behavior", Soc. Pet. Eng. Jrnl., Aug. 1978, p. 265.

Russell, D.G., and N.E. Truitt, "Transient Pressure Behavior in Vertically Fractured Reservoirs", Jrnl. Pet. Tech., Oct. 1964, p. 1159.